Nonparametric Structure Models in Local Protein Conformation Sampling and Design

Alexander Ford

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2018

Reading Committee:

David Baker, Chair

Joseph Hellerstein

Frank Dimaio

Program Authorized to Offer Degree:

Biochemistry

University of Washington

Abstract

Nonparametric Structure Models in Local Protein Conformation Sampling and Design

Alexander Ford

Chair of the Supervisory Committee:
David Baker
Department of Biochemistry

Protein design relies of the identification of a sequence that specifically encodes a target conformation as a folded native state. This native states is encoded by a combination of energetically favorable local and nonlocal structural features. Rapid identification and design of conserved, local structural features may be used to enable and accelerate design of functional or novel non-local interactions. The work here describes the implementation and initial analysis of a turn design strategy based on nearest-neighbor queries against an extensible database of known protein structures. We outline the implementation of this database and search system within the Rosetta biomolecular modeling framework, the successful application of this approach to the atomic-level design of helical hairpin turns and the extension of this approach to combinatorial diversification of a ligand-binding scaffold via a distributed simulation integrating the Rosetta and PyData software stacks.

submitted-0-g231af1f-d_2018-03-11 - Doctoral Dissertation

ALEXANDER FORD

2018-03-11

Contents

1 Nonparametric Structure Models in Local Protein Conformation Sampling and Design

1.1 Introduction

Biomolecular structure modeling is fundamentally based on the exploration of an energy function scoring the relative favorability of molecular states. This state is a joint representation of both chemical composition, in "sequence space", and molecular conformation, in "conformational space". While all biomolecules have dynamic behavior in solution, in this context a "folded" molecule is approximated as having a single native state with a significant free energy gap to other potential states. (*1*) Protein structure prediction and design tasks can then be approached as the identification of the mapping between amino acid sequence space and folded native states in conformation space.

In this approach prediction is an optimization over conformational space given a fixed sequence to identify the native state. In contrast, design is optimization is over the **joint** conformation/sequence space to identify a sequence encoding a target conformation as a native state. The design target may be given as be board description of the target behavior: (e.g., a stable loop (*2*), a fold description (*3*), or a symmetric architecture (*4*)) or a specific, typically atomic-level, representation of a partial solution (e.g., a subset of a protein-protein interface (*5*), a set of ligad-binding residues (*6*), or a theoretical enzymatic state (*7*)). These design requirements may specific a subset of conformational space, as when specifying a target fold, or a subset of both conformational and sequence space, as when specifying the active catalytic conformation of a set of residues.~

Both sequence and conformational space are exceedingly high dimensional. For instance, the sequence space of a 150 amino acid protein is `150^20` (~`33e42`) potential unique sequences. The conformational space of the molecule, assuming fully ideal covalent bond geometry, contains 450 degrees of freedom along the polypeptide backbone alone. A reasonable discretization of this space would encode `10e300` potential conformations.

The sequence-to-native-state mapping is exceptionally sparse in both sequence space and conformation space, as the overwhelming majority of random amino acid sequences do not encode a well-folded protein. Conversely, the overwhelming majority of molecular arrangements, even when considering the subset limited to ideal covalent bond geometry, also do not represent a native state. The space of observed, presumably stable, protein coding sequences, ~100e6 unique sequences in the NCBI RefSeq database (*8*), is a sparse evolutionary sampling of the domain of this mapping. The space of observed structures, ~120e3 protein structures with in the protein data bank (PDB), (*9*) is a **highly** sparse sampling of the of range of this mapping.

1.1.1 Fragments In Sampling

Fragment-based sampling, in which a related set of internal degrees-of-freedom (DOF) derived from an observed structure are sampled as a single step, is a critical component of many conformational sampling algorithms. Restricting sampling to fragments effectively serves as a lower-dimension embedding of protein conformational space and allows sampling within empirically relevant subset of the representable conformations. Moreover, moves within this lower space allow effective search of the exceedingly rough, high-dimensional conformational score landscape by both enabling concerted DOF changes that overcome local score barriers and biasing sampling to conformations more likely to encode score minima.

Many core Rosetta protocols, including de-novo design (*3*), ab-inito structure prediction (*10*) and homology modeling (*11*) rely on fragment-based methods to guide sampling. In these systems conformational samples are selected via a "fragment picking" process, in which the most likely conformations for a given fragment are identified by a sequence-space distance metric. (*12*) These fragments then function as samples for global optimization, typically via Monte Carlo (MC) optimization in a low-resolution model representation followed by refinement with a full-atom score and representation. (*13*) Alternatively, previous work has demonstrated that fragment-based sampling via a conformation-to-conformation distance metric can be used to dramatically increase effectiveness of optimization over the conformational score landscape via sampling relevant local conformational variants. (*14*)

1.1.2 Structure Vs Design

The structure prediction task is based on the observation that a stable sequence encodes a well-folded native state with a significant energy gap to non-native conformations. Any energy function formulation **necessarily** includes degree of inaccuracy, and the Rosetta full-atom energy function has not not consistently modeled the physical energy gap. (*15*) However, expansion and optimization of the energy function with plausible physics-based terms has increased the observed score delta between native conformations and non-native decoys. (*16*) This improvement plausibly suggests that an energy function formulation where the magnitude of error in the energy function less than magnitude of the physical native-like energy gap may exist. With the assumption of a **sufficiently accurate** score function and, critically and empirical evidence that the sequence is has a stable folded state, the prediction task is an exploration of conformational space searching for a single native-like state.

In contrast, the structure design task may be viewed as the joint exploration of sequence space and the local conformational space with the goal of identifying any state with a native like energy gap. This may be performed by optimization of the energy function in this joint space under the simplifying assumption that a deep score minima reflects a native-like energy gap. (*17*) However, this assumption does not reflect the energy distribution, and specifically the possibility of alternate energy minima,

of the designed sequence across the full conformational landscape. Multi-state design strategies, which explicitly address and model this energy landscape, have been used to dramatically increase the effectiveness of designing to a single native-like state. (*18*)

1.1.3 Empirical Search Heuristics & Guiding Design

Design suffers from both the increased dimensionality of the joint structure/sequence space and the inherent sparsity of stable solutions in conformation space. There is always the possibility that no stable solution exists "near" the target designed conformation. However, tools to predict this sparsity may be used in a simplifying assumption to guide design; *many* potential stable sequence/conformation pairs exist, these solutions are unevenly distributed through sequence/conformation space, and the goal is to find any single solution. This assumption has significant implications for the applicability of different sampling strategies in prediction vs design.

A structure prediction tool much be highly sensitive to the possibly of false-negative errors, in which the single correct target sample is filtered from the search, but may be highly tolerant of false-positive errors, in which poorly-performing samples are unnecessarily evaluated. Moreover, the existence of a well-folded native state for the sequence justifies extremely broad and deep sampling of given sequence; the successful structure prediction may require vast sampling to observe the single native-like sample. (*19*)

In contrast, a design tool may be relatively insensitive to false-negative errors, as there are likely many possible solutions to a broadly specified design task. However false-positive errors, in which resources are spent exploring invalid regions of conformation & sequence space, may cause dramatic decreases in efficiency of a multi-step design process. Given this trade off, a strategy that effectively reduces the dimensionality of the search space, even at the expense of discarding potentially valid solutions, can allow the exploration of increasingly complex design tasks.

Exploitation of this error trade off opens the door to many design strategies developed to take advantage of the observed distribution of sequence (*20*) , conformational (*3*) and the joint sequence/conformational space (*21*). All function by restricting sequence or conformational sampling via an empirically observed distribution to a subregion where single-state score optimization is more likely to yield native-like solutions. This distribution may be identified via many strategies, including sequence alignment, structural alignment, or local structural analysis.

Moreover, empirical search heuristics may serve to limit sampling to regions of model state where the model score function has greater effective accuracy. This has been observed in template-based structure prediction (*14*), ab-initio fold design (*3*) and antibody redesign (*22*) as well as many other examples. In these cases the heuristic serves to both focus sampling in more favorable regions of model state and avoid score false-minima generated by lacuna in the energy function.

Despite these obvious potential advantages, practical utilization of a empirical search heuristic to guide design is dependent on the relative cost of evaluating the heuristic vs further unguided sampling. Though it may provide arbitrary improvement in sampling efficiency, if the cost of evaluating the heuristic exceeds the effective reduction in sampling cost then the tool may be practically useless. Pre-evaluation of a static heuristic (eg. homology modeling template selection, ab-initio fragment picking, or structure-base antibody sequence profile generation) may allow amortization of a large cost over many sampling trajectories. However for inline applications, in which the model state is an input to the heuristic, rapid and flexible evaluation is critical.

1.1.4 Existing Approaches to Turn Design

The majority of existing, generic approaches to protein turn design are built over reapplication of structure prediction tools in a design context. (*23*) (*24*) As described above, these modeling tools are developed for identification of conformation given sequence, though application to design typically performs initial sampling via a sequence independent backbone conformational model, and utilize sequence design on individual samples.

These approaches require an computationally expensive, multi-step sampling process to score candidate turn conformations, performing stochastic conformational sampling, optimizing local sequence identity and conformation before ranking candidates via a full-atom score function. Both sampling and discrimination **may** be guided by empirical data, however this is typically used to roughly guide initial sampling toward more favorable regions of conformational space. (*25*)

Reapplication of structure prediction approaches to turn design has significant drawbacks in both sampling and scoring. Though KIC & CCD based modeling **may** be effective in some structure prediction and design contexts, the increased sampling space provided in design severely complicates score-based selection of final models. Selecting designs requires assessment of a large number of models, and the relative magnitude of the native energy gap vs score function inaccuracy is small.

Multistate strategies incorporating both negative and positive design, in which a conformational ensemble is used to assess the favorability of a target sequence, has proven to be effective at denovo design of specific loop conformations.However, these approaches are extremely computationally intensive and infeasible for use in early stages of a complex multistage design pipeline. (*26*)

In many design tasks the target turn conformation and sequence are unconstrained; the only requirement is that a turn accommodate the design constraints satisfied by the rest of the target structure. If the designed turn sequence adopts a specific local conformational this may favor the overall designed state by disfavoring alternate states, a form of implicit negative design. Additionally, conserved turn features may have a strong effect on folding rate and fold stability. Conversely, if the turn sequence is energetically disfavored this may cause misfolding or prevent folding, regardless of the favorability of

the remainder of the design. (*27*) Deconvolution of these effects may dramatically increase difficulty of analysis of many low-throughput design experiments, in which a relatively limited number of designs may be experimentally characterized.

Observed protein structures contain a variety of recurrent, sequence associated structural motifs (e.g., beta hairpins, alpha-beta turns, helical capping residues, and helix hairpins) forming a wide variety of turn solutions joining secondary structure elements and exhaustive analysis of poly-peptide conformational space provides strong evidence that a discrete number of allowable inter-secondary-structure turn conformations may exist. (*28*) (*29*) Though the observed coverage of available local conformational space in the PDB is incomplete, there is significant coverage in the length ranges represented by many turn motifs. (*30*) (*31*) Moreover, empirically derived sequence information may be used to dramatically increase the reliably and stability of designed proteins, particularly when used in conjunction with energy-function guided sequence design (*32*)

Taken together, the requirement of high confidence in turn design and known consistency in observed turn structures suggests an essentially conservative approach: only design turns for which there is strong empirical evidence of a locally favorable conformation and sequence encoding that conformation. Previous work in denovo design enforced this restriction by biasing fragment-based sampling around strongly conserved turn features, and then utilized full-atom refinement to assess the favorability of sampled tertiary interactions. (*3*) However alternate design tasks may necessary begin with the design of tertiary structure, requiring the subsequent design of highly favorable local features compatible with these pre-identified non-local interactions. While this may be accomplished via post-design filters of design generated via any of the methods described above, this conservative strategy allows optimized sampling strategies focused on delivering high reliability and low computational cost.

1.1.5 Structural Nearest-Neighbor Models for Turn Design

The work here describes the implementation and initial analysis of a structural nearest-neighbor search strategy developed to support the rapid identification and design of highly optimal, contiguous protein structural fragments. This local search strategy is implemented via a layered software stack of independent modules supporting multiple, interdependent conformation and sequence search heuristics via a shared structure/sequence database.

The local design strategy is applied to the diversification of a designed protein-protein interface via turn resampling & redesign in context with the binding target.

The strategy is applied to the specific task of designing conserved turn conformations to join parametrically designed secondary structure elements. The work then extends and generalizes this local design strategy into a combinatorial search framework for the design of multiple, potentially interacting, turn segments from a discontiguous starting structure.

Finally, the combinatorial design strategy is applied to the task of redesign of a ligand-binding beta-barrel, generating a 45,000 member combinatorial design library to allow robust evaluation of both the local and combinatorial search strategies.

1.2 Implementation

This set of related protocols is implemented via a set of modules spanning the complete Rosetta software stack, from low-level numeric operations to high-level support for distributed computation. This decomposition is intended to support flexible reapplication of each layer to novel protocol development, as well as facilitating rapid prototyping and interactive analysis. The core implementation is available via the RosettaCommons `main` repository, with protocol component implementations available as dependent external repositories.

Component	Implementation Location
Core Structure Database and Query Components	RosettaCommons/main#2765
`pyrosetta.distributed` Support	RosettaCommons/main#2760
Prototype Combinatorial Design	asford/interface_fragment_matching
Beta Barrel Combinatorial Redesign	asford/av9_loop_rebuild

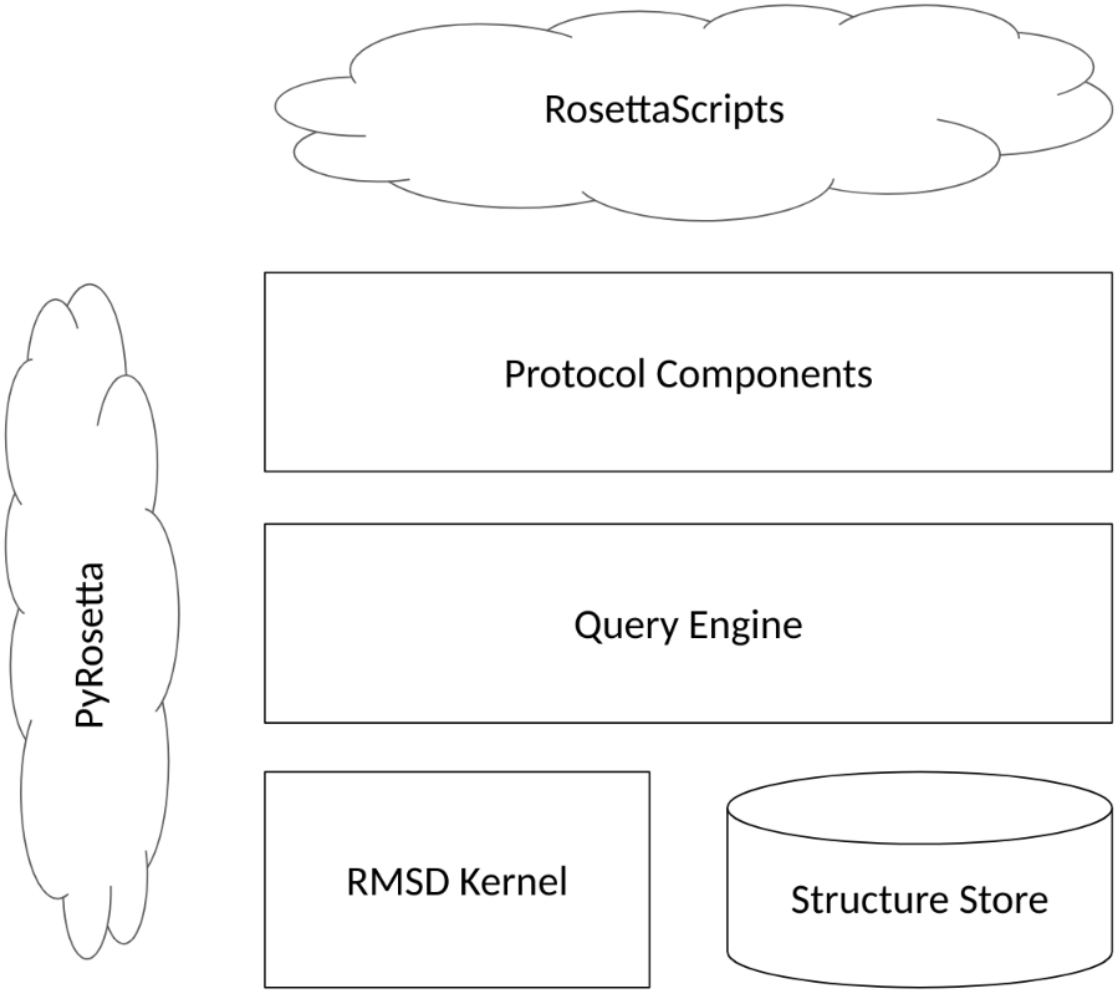

Figure 1: Structure store implementation architecture. RMSD kernel implemented in the `rosetta::numeric` module level and `pyrosetta.core.rmsd_calc` namespace. Structure store in the `rosetta::core::indexed_structure_store` level and `pyrosetta.core.structure_search` namespace. Protocol components in the `rosetta::protocols::indexed_structure_store` module level.

1.2.1 Numeric

The fundamental numeric primitive supporting this framework is an implementation of the QCP Superposition method, providing support for rapid calculation of aligned all-atom coordinate RMSD and the superposition transform. (*33*) This algorithm provides two fundamental advantages over existing coordinate superposition algorithms: a significant increase in runtime performance over existing superposition algorithms when applied to small coordinate systems and a further increase

performance when the only the RMSD distance metric, rather than the full superposition transform, is required.

To support integration within the larger framework the algorithm is reimplemented as a header-only C++ template using the Eigen numeric library. (*34*) This implementation provides simple support for zero-copy evaluation of the superposition algorithm over existing coordinate representations, allowing efficient execution without heap memory allocation. Use of the Eigen Geometry modules allows rapid calculation of the superposition as a homogeneous transform matrix given the rotation matrix returned by the QCP algorithm.

This kernel is then used to implement a collection of zero-copy, pure functions for calculation of RMSD alignment metrics over buffers of coordinate data, with specific support for broadcast and pairwise alignment of coordinate sets. To facilitate varying input data formats, this implementation provides support for input coordinate data in dense or arbitrarily strided arrays as well indexed coordinate buffers. A zero-copy numpy-compatible interface is provided via the pybind11. (*35*)

1.2.2 Core Structure Store

The core data model for the structure store is a partially redundant representation of protein sequence and conformation facilitating generic queries between structure and sequence space. The data model contains a complete description of the protein internal bond torsions, sequence identity, linear connectivity and backbone cartesian coordinates. This representation allows full description of backbone heavy atom bond geometry, but is limited to describing sidechain rotameric identity. Ideal side chain atomic coordinates may be recovered via forward kinematics. Non-ideal side chain coordinates of the closest score minima may be trivially recovered via cartesian or internal coordinate minimization.

Forgoing a cartesian coordinate representation of side chain heavy atoms has the distinct advantage of supporting a fixed-width data format. This allows efficient representation of the database as C++ POD types & numpy structured dtypes, zero-copy buffer-based interconversion between these data formats, flexible data persistence & serialization and greatly simplified informatics & interactive analysis via standard python data analysis tools.

The set of structures identified in the Rosetta 2011 `vall` database (*12*) were pre-processed via coordinate-constrained `FastRelax` under the `talaris2014` energy function and stored in both `hdf5` and `json` format. This data store contains approximately `4e6` residue entries drawn from `16e3` unique protein crystallographic structures pruned for structure data quality and sequence homology. This source database was used for all further analysis and design.

Beyond the database described here, this format has proven useful in managing protein structure data in such tasks as: analysis of decoy collections in structure prediction tasks, storage of large collections

of protein side chain and fragment docking results, and data management for distributed protein modeling and design tasks.

Structure data access is managed via a generic `StructureStoreManager` interface, responsible for retrieving and caching the core data model from an arbitrary backing store. The core Rosetta implementation provides backing store implementations for initializing a structure database from: pdb-formatted files, `silent`-formatted files, `hdf5` database files and human-readable `json` record files. These data types provide a spectrum of flexibility/efficiency trade-offs. pdb/`silent` record files provide maximum flexibility with a significant read-time cost to render the database format, `hdf5` files provide maximum efficiency but require database preprocessing, and compressed `json` records provide a reasonably performant, compact storage format amiable to manipulation via standard tools.

To ensure forward-compatibility and accommodate specialized applications, the pyrosetta structure database interface was extended to support dynamic registration of structure backend providers. Registered providers utilize the python buffer-interface to support zero-copy data transfer between the dynamic provider and structure store cache. A dynamic provider supporting access to HDF5 files via `h5py`, rather than the Rosetta HDF5 build variant, is provided in the standard pyrosetta distribution and allows simplified access to `hdf5`-formatted databases using the efficient `blosc` compression filter. This interface may be used to provide support for databases accessed via a remote resource, shared among multiple processes via shared memory segments, or stored in alternative record formats.

1.2.3 Core Structure Queries

The core nearest-neighbor model is implemented as an specialized coordinate search database supporting optimized execution of multiple query types. Previous Rosetta database formats relied on exhaustive pre-indexing of source data to support rapid geometric query evaluation, resulting in either (a) significant growth in database size to accommodate extensive pairwise indices or (b) severe restrictions on the available queries to a consuming application. In contrast, this database implementation relies on on-demand indexing of the source structure store followed by query execution.

The database supports two core query types: a contiguous backbone coordinate fragment alignment (`StructureSingleQuery`) and a discontinuous backbone coordinate fragment alignment (`StructurePairQuery`). Both queries are specified by a set of query fragment length, target backbone atomic identities, target backbone atomic coordinates and an alignment RMSD tolerance. Each query result is an array of indices in the source database indicating the fragment start point and alignment RMSD. The consuming application is then expected to utilize these indices to retrieve any required information from the source structure store.

The contiguous query is trivially implemented as set of fragment-fragment RMSD calculations and scales on the order of O(N) for a structure database of size N. The pair query is conceptually a set of

pairwise comparisons against all possible pairs of fragments within each structure within the database and scales on the order of O(N * L^2) for a database of size N and average structure size within the database L. As the average length of a structure is on the order of ~100 residues a naive implementation of the pair query will execute four orders of magnitude slower than a single query.

All query classes utilize an on-demand fragment index cache to optimize query execution at the specified fragment length. To calculate the minimum inter-fragment RMSD the QCP algorithm requires (a) the center of mass (COM) of each coordinate set, (b) the inner product of the two, centered coordinate sets and (c) the first eigenvalue of a key matrix. Initial benchmarking demonstrated that calculation of the fragment center of mass represented a significant fraction of the alignment runtime. The cached index stores the start index and center of mass for every valid contiguous structure fragment of a given length within database, amortizing the cost of this calculation across repeated queries.

The structure pair query is optimized by two bounding criteria: an optional limit on the inter-fragment sequence distance and an implicit bound on the inter-fragment center of mass RMSD determined by the query RMSD tolerance.

The sequence distance cutoff is commonly used in segment sampling protocols and, if present, only necessitates comparison against a fixed number of fragments allowing query execution O(N).

The pair query inter-fragment center of mass cutoff is derived from an analysis of minimum bound properties of the all-atom fragment RMSD tolerance. (*36*) In order to rapidly query for fragment centers with a given distance a secondary RTree index over the fragment center of mass cache is generated for each structure, allowing effectively constant-time identification of the subset of fragment pairs that may match the given query. (*37*) While this does not provide an assurance an O(N) scaling, particularly in cases where the query RMSD tolerance is large (eg. >3A) in the majority of practical applications is also reduces the effective query cost to a constant multiple of the database size.

1.2.4 Protocol Components

Protocol-level components are implemented as Rosetta “Mover” components, integrated into the RosettaScripts scripting framework. (*38*) Briefly, this framework operates on complete molecular models, termed Poses, and describes protocols as a set of transformations applied to Poses, termed Movers, and a set of scoring components, termed Filters. Movers may be subdivided into two types, those generating a zero-or-one output Poses from a given input and those generating zero-or-more output Poses.

Conformational sampling is implemented via a multiple-output `DirectSegmentLookupMover`, which combines a database query, cluster analysis of the lookup results, and generation of valid output models. The protocol component is intended to produce a diverse set of candidate models ranked by approximate local structure density.

The mover receives as input a source pose containing a segment with disconnected chemical connectivity represented as a pose "chain break". It produces as output zero-or-more molecular models of lookup results superimposed and joined to the input structure. Critically, it produces models with valid chemical connectivity but does not produce models with valid bond geometry. Generation of "valid" models requires downstream minimization or sampling via an additional protocol.

The `DirectSegmentLookupMover` proceeds by first executing a `StructurePairQuery` on two fragments adjacent to the target chainbreak, using the specified input fragment length and query rmsd tolerance. Matching contiguous fragments identified by this query are then extracted from the source structure store and coordinate superimposed onto the endpoint fragments. This result set is then clustered by fragment-length and unaligned fragment coordinate RMSD, producing a set of candidate samples with conformational diversity greater than the specific alignment threshold. Samples are then optionally pruned by conformational sample density by removing samples with a cluster size below a provided threshold. Finally samples are output in order of decreasing sample cluster size.

To facilitate downstream analysis, the mover component annotates output poses with residue label metadata indicating the regions modified via the segment lookup. This allows recovery of the inserted segment as a residue selection in downstream protocol components and is used in the described design protocols to restrict minimization to the inserted region, determine if the target region produced viable bond geometry post-minimization and restrict sequence profiling and design to the inserted region.

Sequence sampling is implemented via a single-output "SegmentSequenceProfileMover", which combines a database query, sequence profile analysis of the lookup results, and application of a sequence profile constraint to the target pose. The protocol component is intended to generate a structure-derived sequence profile for a contiguous local segment to guide downstream design.

The mover receives as input a source pose and a residue-selector based specification of the target segment. It produces as output a molecular model with sequence profile constraints representing the point-specific log-odds deviation from background amino acid composition applied to the target region. These constraints may be used in downstream design via a modified full-atom score function.

The `SegmentSequenceProfileMover` proceeds first executing a `StructureSingleQuery` on the single, contiguous structure fragment indicated by the target selection using the specified query rmsd tolerance. The amino acid sequences of matching fragments are then extracted from the source structure store and used to generated a position specific observed sequence count. This count is then pseudo-counted by a multiple of the observed background amino acid composition. Finally, a point-specific scoring matrix of the log-odds deviation from the observed background amino acid composition is generated for the segment and applied via profile sequence constraints.

The pseudo-count and log-odds strategy utilized in the mover is intended to limit the level of score double-counting against existing score terms including amino acid reference weights `ref`, Ramachan-

dran potentials `rama` & `omega` and phi-psi specific amino acid propensities `p_aa_pp`. This strategy is likely insufficient to eliminate all double-counting effects, as it is a conformation-independent correction unlike the conformation dependent `rama`, `omega` and `p_aa_pp` terms.

1.2.5 Design Protocols

Effective sequence design and design selection relies on identifying a local score minima in both sequence and conformational space fulfilling design requirements. Given the heterogeneity of design tasks and broad applicability of the conformational sampling and sequence profiling protocols these subsequent steps are decoupled from the components described above and typically executed via the RosettaScripts framework. In the results presented here minimization and sequence design were performed by a common protocol combining constrained torsion-space or cartesian-space backbone minimization, fixed-backbone packing-based sequence design or flexible-backbone relaxed-based design and score-based design selection.

Minimization is used to resolve bonded models from candidate fragments identified via the `DirectSegmentLookupMover`. The candidate structure is converted to a partially sequence free backbone representation in which all fragment residues other than glycine and proline are converted to alanine to represent a neutral scoring background. Glycine and proline residues are preserved as these residues encode unique local structural features with backbone torsional propensities dramatically different than the remaining amino acids.

The initial sampling protocol implementation utilized a constrained torsion-space minimization component to resolve bonded models. The poly-ala segment backbone model was extracted into a “minimization pose”, containing the segment rooted to a single virtual atom in the aligned reference frame. Harmonic cartesian coordinate constraints with .1A standard deviation were added to the aligned segment endpoints and constraints with 1A standard deviation were added to the segment backbone coordinates. The segment internal torsions rigid body orientation was then minimized using the standard Rosetta `MinMover` under the constrained, default non-cartesian full atom score function. Models which failed minimization or who’s local minima resulted in non-ideal cartesian bond geometry are discarded.

The primary sampling protocol implementation instead minimizes the segment alignment via in-context cartesian-space minimization. The segment structure is directly inserted into the target model generating a closed model containing at least two sections of non-ideal bond geometry at the segment insertion points. The inserted segment is then subjected to cartesian minimization using the standard Rosetta `MinMover` under the default cartesian full atom score function, which allows concerted changes in the inserted to structure to adopt near-ideal bond geometry. As previously implemented, models which fail minimization or who’s local minima do not result in near-ideal cartesian bond geometry are discarded.

Following minimization the design segment is profiled via the `SegmentSequenceProfileMover` using a profile distance roughly equivalent to that used in lookup clustering, generating a set of sequence profile constraints for the segment and a count of structures within the threshold distance. Models with a small number of structural neighbors (defaulting to 10) post-minimization are discarded.

Taken together, cartesian minimization and re-profiling serve to identify the subset of samples in which the RMSD alignment distance metric accurately identifies a designable neighbor conformation. Bond-geometry filtering serves to eliminate “false positive” cases in alignment distance does not accurately reflect the more complex conformational distance and realignment serves to eliminates samples which must be dramatically modified to accommodate the target endpoints.

In the typical structure design tasks, in which the goal is to identify stable but non-functionally restricted local designs, the collection of candidate segments are designed via a fixed-backbone packing protocol utilizing the generated structure-based profile. Each candidate undergoes a single round of side-chain packing followed by side-chain torsional minimization, generating a final segment design. Candidates are then ranked by mean, unconstrained full-atom score per residue across the designed segment. Unconstrained score is used to identify candidates representing local score minima with high likelihood of recovery in downstream design protocols and most likely to represent detection of a valid structural feature.

In sampling protocols requiring multiple candidates per segment location, such as loop diversification or combinatorial library generation, clustering and score-based selection is used to identify a structurally diverse set of samples from the candidate pool. Hierarchical agglomerative clustering with average-linkage over fragment RSMD is used to generate a set of flat clusters at either (a) a specified minimum clustering linkage tolerance or (b) maximum number of candidate samples. In candidate pools with variable-length members, fragments with differing length are assigned an effectively infinite inter-sample distance. The single best-scoring sample from each flat cluster is then passed through to the following protocol steps.

[Figure inline design protocol.]

1.2.6 Combinatorial Design

Many tasks necessarily require sampling or design of multiple structual segments, resulting in a combinatorial design task. This may occur when designing multiple ideal closures through an arbitrary core structure, such as the design of closed structures from parametrically defined helical bundles or beta barrels. Alternatively, this may occur when designing defined functionalized segments of a core fold, such as the CDR regions of the IG fold or active site loops of the TIM barrel fold.

To address this class of related problems we opted to develop a generalized framework for multi-segment design tasks. The framework uses a simple dynamic programming strategy to decompose the

full task into multiple individual segment closure sub-tasks, identify subtask solutions that may form components of a full solution and recombine these sub-solutions to identify and select final designs.

The full task, termed “topology closure”, can be framed as designing a path though n disconnected structural segments by linking segments with designed closures with varying restrictions on segment ordering. In the unrestricted case there are n! potential element orderings, composed of combinations of n^2 pairwise segment closure sub-solutions. In the “circulation permutation” case there are n potential element orderings, composed of n+1 segment closure sub-solution. Finally, in the “fixed ordering” case there is a single element ordering composed of n segment closure sub-solutions.

For example, in unconstrained parametric helical bundle design n helical secondary structure elements are provided in a specific geometric arrangement where some subset of the helical endpoints may be joined with a short canonical turns. The solutions are then a set of helical element orderings and the set of designed turns joining those elements in the given ordering. In the “antibody design” framing, the framework region may partitioned into disconnected segments via “excision” of the CDR regions and then recombined in the same ordering by redesigned CDRs.

For “trivial” unconstrained closure tasks, such as the closure of 2-3 structural elements, a simple enumerative sampling approach may be sufficient, however the n! scaling over potential path length and n^2 scaling over sub tasks presents an obvious barrier as the problem size grows. To optimize the search process the closure component uses a simple directed graph formulation to reuse common sub-solutions between varying closure paths and perform dynamic pruning of unneeded sub-tasks.

In the search graph each source element is considered a vertex and potential inter-element closure an edge. The search begins by performing exhaustive inter-element segment lookup, defining a sparse graph in which the directional edge `i,j` exists iff a candidate solution was found between nodes `i` and `j`. A simple traversal of the graph is executed beginning from every node, identifying all unbranched spanning paths through the source graph. The connectivity graph is then pruned, removing all inter-element paths that are not members of a spanning path. Each remaining directional edge is then refined via the complete design protocol to generate the final closure sub-solution. Finally each spanning solution is generated and refined via recombination and design of the component edge sub-solutions.

To enable “fixed ordering” and “circular permutation” within the same framework, the topology closure implementation allows restriction of the initial connectivity search to arbitrary subset of graph edges. For fixed ordering this is includes edges `(i, i+1)` for `i` in `0..n-1`. For circular permutation this includes edges `(i, i+1)` for `i` in `0..n-1` and `(n, 0)`. More complex closure scenarios, such as arbitrary permutation of loop connectivity while preserving termini locations, may be specified by more complex edge subsets. The traversal restrictions described above then trivially restrict the closure search process to the desired element orderings.

While the optimization described above may be ineffective for arbitrarily connected graphs, we observed that the geometric constraints of the topology closure task yield a relatively sparse initial

connectivity graph, particularly when specifying short, ideal closure segments. This enables very rapid enumeration of the small number of available spanning closure paths and dramatically reduces the effective search space. Notably, this observation may fail to hold if closure length and quality restrictions are relaxed.

The computational cost of design of fully connected backbone decoys is highly dependent on the degree of interaction between the identified closure segments. In instances with independent solutions, as occurs with short turn designs, the validity of a sub-task solution is a strong indicator of it's overall contribution to the final task score. In this context a small number of ideal sub-solutions is preferred. In instances with interdependent solutions, as occurs with longer loop designs or designs with significant spatial constraints, each sub-solution must be valid, but sub-solutions be mutually incompatible. In this context a larger collection of sub-solutions must be identified in order to search for a mutually compatible end-solution.

The sampling depth of each sub-task as well as the sampling depth of each closed solution is left as a configurable parameter to accommodate application specific tuning. In instances where relatively little diversity is required the use of restrictive lookup constraints and a shallow sub-task search depth allows exhaustive enumeration of the available closure solutions for each closed path. In instances requiring greater structural diversity a deeper sub-task search depth may be used to generated a larger collection of potential closed solutions per path, which is then randomly sampled to generate designed closed solutions.

1.2.7 PyRosetta Distributed Interface

To support an effective task-based distributed solution for these disparately scoped tasks we developed a generic integration layer between between rosetta-based modeling protocols and the broader python scientific computing stack. Integration is provided via a set of distributable primitives encompassing model state, in the form of Rosetta poses, and model actions, in the form of RosettaScripts protocols. Taken together these components provide a basic data types and functional forms for molecular modeling via a wide variety of standard python libraries.

The standard python ecosystem relies on a small set of core interfaces to support interoperability between independent library components. Critically, the vast majority of python distributed computing libraries utilize the `pickle` interface to provide transparent serialization-based distribution. To support this interface the `pyrosetta.distributed` namespace implements a set of high-level components utilizing the `pyrosetta` API to manage instantiation and initialization of underlying compiled Rosetta components.

The Rosetta `Pose` object, a fundamental datatype including a full molecular system and it's associated metadata, is represented by the `PackedPose` interface. The `PackedPose` contains a serialized

`Pose` object as an opaque, compressed buffer as well a read-only mapping of scores extracted from the pose object before serialization. These scores include all active components of the Rosetta score function most recently applied to the Pose as well as scores reported by all filters previously applied to the Pose. Critically, all terms reported within the packed pose may be recovered via deserialization of the underlying Pose buffer, which functions as the "primary" data representation.

RosettaScripts protocol components are represented via a set of task objects containing an XML encoded protocol specification. Two task classes are provided, a `SingleOutputRosettaScriptsTask` and `MultipleOutputRosettaScriptsTask` respectively defining either a one-to-one component returning a single output or one-to-many protocol component returning a python `generator` of outputs. Task components operate on a `PackedPose` representation and function by (1) initializing a RosettaScripts protocol object from the source XML script, (2) deserializing the input `PackedPose` into a short-lived `Pose` object, (3) applying the parsed protocol to the `Pose` and (4) serializing the resulting model as a `PackedPose`. Task objects use a simple caching strategy to reuse the results of XML parsing and XSD validation of the RosettaScripts component, amortizing this relatively expensive operation over many protocol applications.

To facilitate interactive data analysis and integration with the larger python software stack `pyrosetta.distributed` includes a set of adapter functions defining isomorphic mappings between Rosetta `Pose` objects, `PackedPose` objects, byte-buffers of serialized poses, a `dict`-based representation including both scores & `b64` encoded pose buffers and a `pandas` DataFrame of scores and pose buffers. These basic representations allow streamlined interaction with existing analysis libraries (eg. `pandas` statistical analysis, `sklearn` model fitting and application, etc...) and integration with standard storage formats (eg. `json`-encoded text files, `avro` record-oriented storage, `parquet` column-oriented storage, etc...). Functions and tasks within the `pyrosetta.distributed` namespace utilize these adapter functions to provide a type-agnostic external interface capable of operating on any of the supported data formats.

By virtue of reliance on standard python primitives the `pyrosetta.distributed` namespace is not tightly coupled to a single execution engine. Single-node scheduling may be managed via the standard `multiprocessing` or `concurrent.futures` interfaces, providing a zero-dependency solution for small-scale sampling or analysis tasks. Existing commodity clusters may be managed via the `mpi4py` interface, providing integration with existing high performance computing scheduling and job frameworks. Cloud-based infrastructure may be managed via `dask.distributed` and `daskernetes`, providing a portable, scalable solution for modern compute environments.

[Figure Rough class diagram of `pyrosetta.distributed`]

1.2.8 Distributed Combinatorial Design via Dask

In contrast to many Rosetta-based modeling tasks, the combinatorial search strategy outlined can not be described as a set of fully independent sampling trajectories. While the most compute-intensive phases of the calculation, individual segment design and closed structure design, are trivially-parallelizable, the closed-path identification and closed structure combination steps require recombination of multiple sub-solutions.

The sampling requirements of different design tasks vary dramatically, and may not be predictable for a given task. For instance, in parametric design tasks the objective may be to computationally screen candidate backbone solutions and return a single closed result requiring relatively little sampling per task. This may be simply accomplished on a single node, with no intercommunication between parametric samples. In contrast, a combinatorial library design task may require much deeper sampling, including generation of hundreds of solutions per sub-task and sampling thousands to hundreds of thousands of closed model candidates. In these cases a single node is clearly insufficient and a distributed solution requiring the coordination of hundreds to thousands of cores may be required.

To facilitate a shared solution between these disparately scoped design goals the combinatorial search process is implemented as a parallelizable dataflow graph via `dask`, a dynamic task scheduling library. (*39*) In this programming model the search process is described as a collection of independent units of defined input and output arranged as a directed acyclic task graph. An external scheduling process then guides execution of the graph via pool of worker processes managing data storage, task data dependencies and task scheduling. Utilization of the `dask.distributed` scheduler allows execution of the combinatorial search protocol in systems spanning single-process execution to distributed systems of dozens of nodes with thousands of cores.

While the distributed search process is limited to thousands of concurrent tasks in the current implementation, this is unlikely to present a scaling barrier in most relevant design cases. The combinatorial process requires arbitrary inter-node communication for each pruning stage, however each top-level topology closure task group is entirely independent. In the vast majority of design processes multiple independent topology closure tasks are required, allowing simple horizontal scaling via the use of multiple schedulers and worker pools.

High-level parallelism also allows for efficient compute resource utilization despite several intrinsically serial phases of the closure search process. Execution of multiple independent closure groups on a shared worker pool allows dynamic reallocation of workers between independent task graphs, therefor reduced potential parallelism in one graph simply allows expanded concurrent execution of another task group. The runtime of serial phase of the topology closure search is dwarfed by the trivially parallel design tasks, and concurrent execution of two or more closure tasks has proven to be sufficient to achieve complete utilization.

1.3 Applications and Results

1.3.1 Coordinate Alignment versus Geometric Hashing

To provide an initial test of primary feature recovery via conformation-guided sampling we utilized a benchmark test set of de-novo designed proteins with resolved NMR structural models. (*3*) We generated a test cases by extracting all contiguous spans of non helix-or-sheet identified residues within the source structures and excised these spans from the source structures, leaving a set of disconnected secondary structure elements. As this validation set was generated via recombination of highly-conserved local structural features we hypothesized that any non-locally guided sampling system must be capable of recapitulating both the designed and experimentally observed conformations. To quantify local segment recapitulation we calculated backbone heavy atom rmsd between the source feature and all candidate conformational samples, and considered a turn "recovered" if it included a sample within 1A backbone heavy atom RSMD.

To benchmark against available tools we assessed recovery of turn conformations via a geometric hashing algorithm used in the Rosetta Matcher (*40*) and LoopHash (*14*) sampling algorithms. Briefly, the geometric hashing process defines a characteristic inter-residue rigid body transform and defines a 6 dimensional discretization of the transform via 3 translational DOFs at ~2A bin width in a cartesian representation and 3 orientation DOFs in an euler angle representation at ~25 deg bin width. The sampling process calculates the hash value for the segment to be sampled and then compares this value to hash value of every 3-9 residue segment present in a target database, returning backbone torsion representation of all local segments with the same hash bin.

To assess the efficacy of non-hash-based lookup we benchmarked recapitulation performance via the backbone coordinate RMSD based search algorithm described above, using an endpoint lookup tolerance of .6A and 1A backbone heavy atom rmsd. Samples were drawn from the 2011 `vall` database as previously described.

To assess the relative performance of stochastic sampling approaches we generated candidate samples via both perturbation and unbiased sampling followed by kinematic closure. (*24*) Perturbed kinematic samples were generated with the Rosetta 3.4 application `loopmodel` via `-loop:vicinity_sampling`, performing sampling via perturbation of the target turn conformation backbone degrees of freedom. Unbiased kinematic samples were generated via `-loop:remodel`, performing sampling via stochastic sampling of the Ramachandran distribution of the native turn sequence. Both protocols were used to generate 150 successful closure solutions.

The transform-hash based lookup system fails to resample **any** local segment conformation a non-trivial number of benchmark test cases, though in instances where it did identify samples these closely matched the test case conformations. Furthermore, lookup results, and corresponding failures, are not rotationally invariant, resulting in inconsistent recapitulation results depending on the source

orientation.

These recapitulation failures are a result of hash discretization in both rotational and translational space, but are exacerbated due to rotational binning via an euler angle representation resulting in highly-uneven bin sizes and bin boundary distances, particularly in pole regions of the representation. (*41*)

In contrast, the endpoint alignment sampling protocol recapitulated the turn regions of all design models with a lookup endpoint tolerance of .6A and the turn regions of all experimental models with a lookup endpoint tolerance of 1A.

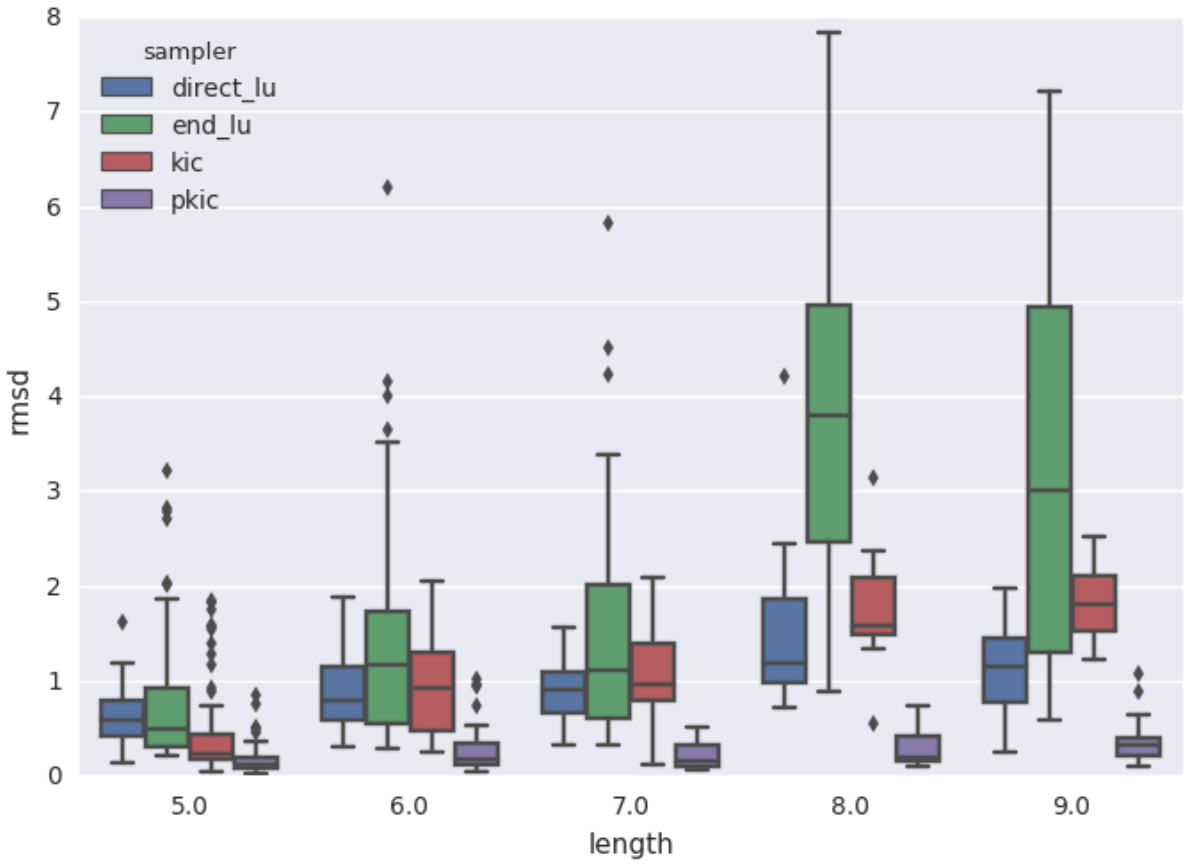

Figure 2: Native benchmark recapitulation performance by summed turn and endpoint fragment length. `pkic`: Perturbed kinematic sampling. `kic`: Unbiased kinematic sampling. `end_lu`: Rigid body transform hash sampling. `direct_lu`: Endpoint RMSD alignment sampling. Transform hashing fails to sample ideal turn conformations in a length-dependent manner, while endpoint alignment and stochastic sampling recover ideal turn conformations.

Test Case	Design Recap Rate	NMR Model Recap Rate
LoopHash Lookup (Source Orientation)	%70 +/- 5	%60 +/- 5
Resampled LoopHash Lookup (10 Rotation Samples)	%95 +/- 3	%85 +/- 5
Endpoint RMSD Lookup (.6A)	%100	%75 +/- 5
5 Endpoint RMSD Lookup (1A)	%100	%100

1.3.2 Coordinate Alignment Throughput

The sampling and profiling protocols are intended for use as sub-components of larger multi-step structure prediction and design protocols. As such, though the coordinate alignment based database search is trivially parallelizable, we assessed singled-threaded database load and query performance.

The two primary structure store backing formats are a `gzip`-compressed json record format and a `blosc`-compressed `hdf5` data store. As the structure database is cached on a per-process basis, the runtime cost of reading the store is amortized over many sampling trajectories. When reading from local, ssd-backed storage, the wall-time of loading a `json`-formatted `vall` store was recorded at `45+-5` seconds and a `hdf5`-formatted `vall` store at `5+-1` seconds.

The RMSD kernel compute time is dominated by root-finding stage of the NR-QCP algorithm for systems of less than 40-50 atoms. As the relevant neighbor models are limited to backbone heavy atoms, the algorithm runs in approximately constant time per fragment for fragments of less than 10-12 residues. For these systems the effective alignment rate is `2e6` alignments per second, resulting is `2.5` seconds per query for contiguous fragments against the `vall` database.

Scaling of discontiguous fragment queries depends on either (a) restriction of the pair query via sequence-space distance limitation or (b) optimization via the fragment center-of-mass alignment threshold distance. In the `DirectSegmentLookup` protocol component described here the default segment length constraint of 6-10 inserted residues limits the sequence space distance between the query fragment start points to 4-8 residues, and the query-per-second rate scales linearly to `20` seconds per query. The performance of unconstrained queries is significantly less consistent, and is dependent on the query threshold distance and spatial relationship between the query endpoint. On the presented native recovery benchmark set pair query times were within the range of `30+-10` seconds per query.

1.3.3 Helical Bundle Hairpin Design

The prototype described above was use to generated turn designs for a collection of parametrically described designed helical bundles, assessing efficacy in design of conserved, local structural features

compatible with designed tertiary interactions.

A collection of symmetric helical bundle architectures, previously described, were generated and screened for compatibility with fully-satisfied core hydrogen bond networks via the HBNet design protocol. Backbone architectures with successful hydrogen bond network models were rapidly screened for connectivity via the closure protocol described above. (*42*)

To identify short “helical hairpin” features the segment lookup was limited to a small endpoint RMSD tolerance (.75A) and short insertion length (<8 residues). To facilitate rapid screening of a large pool of parametric designs the structure library was pre-pruned to a set of loop structures, as identified by DSSP sequence pattern (`[HE]{1,5}L{1,6}[HE]{1,5}` in POSIX extended regular expression syntax). The described torsion-space minimization protocol and sequence design protocols were then used to select the single best-scoring design candidate conformation and sequence at each closure location. The loop closure application used for symmetric, combinatorial closure is available as a standalone application at https://github.com/asford/interface_fragment_maching@3f391e6.

Given the previously identified anomalous stability of designed helical bundles we postulated that non-local core interactions were the overwhelming determinate of design stability. This hypothesis is supported by the observation that bundles can obtain significant stability from designed non-local interactions even in the presence of highly disfavorable turn sequences, as the bundle repeat architecture allows for the effect the turn conformation to rapidly “dissipate”. Unwinding and distortion may effect the terminal repeats, but favorable interactions in “internal” repeats drive adoption of the modeled conformation. (*43*)

After initial characterization, ten designed bundles were crystalized and analyzed for agreement between the designed model and resolved crystal structure. In the overwhelming majority of designs designed helical hairpin conformation recovered in the crystal structure, resulting in low backbone coordinate RMSD.

Source	Turn Backbone RMSD	Turn Sequence
2L4HC2_24 (5J10)	0.743623	QEDPSDE
2L6HC3_13 (5J0H)	0.164282	EKNPSED
2L8HC4_12 (5JQZ)	1.44491	SRGDTE
2L4HC2_23 (5J0K)	1.46539	REGSSDE
3L6HC2_2	1.31627	KQGASEK
3L6HC2_2	1.08268	RSSSSSR
2L4HC2_9 (5J73)	0.336005	KRGVSSD
2L4HC2_11 (5J2L)	0.807964	DERTSTAD

Source	Turn Backbone RMSD	Turn Sequence
2L6HC3_6 (5J0J)	1.11491	EKNPDKD
5L6HC3_1 (5IZS)	0.241546	SELTDEK
2L6HC3_12 (5J0I)	0.321633	KKNPSED

1.3.4 Beta Barrel Combinatorial Redesign

To generate a robust test of the combinatorial design pipeline described above we performed loop diversification and redesign of a designed protein binding the fluorophore DFHBI, HBI32, generating a multi-component library for high throughput stability screening. DFHBI is fluorescent **only** when bound and restricted to a planar conformation, causing fluorescent signal to be dependent on both scaffold folding & ligand binding and allowing screening via yeast surface display. Additionally, designs may be screened via an established high-throughput protease-resistance assay, allowing an alternate measure of scaffold stability. (*44*)

The generated library functions as both an end-to-end test of the integrated pipeline and internal test of pipeline components by combining design and control pools covering the single-loop diversification protocol, the combinatorial diversification protocol and sequence design components.

HBI32 is a beta-barrel scaffold containing a DFHBI binding site capped by four loops. The binding interface is formed by a set residues in the interior of the barrel, however the loop regions both contribute to direct interaction with ligand and stabilization of the overall fold. As with other beta topologies the barrel is tolerant to extensive modification of the loop region, with stable structures spanning short hairpin turns to complex interdependent loops. Previous redesign of the loop regions has demonstrated that variation within the region **may** result in increased fluorescence, however many redesigned constructs result in reduced fluorescence despite being folded monomers.

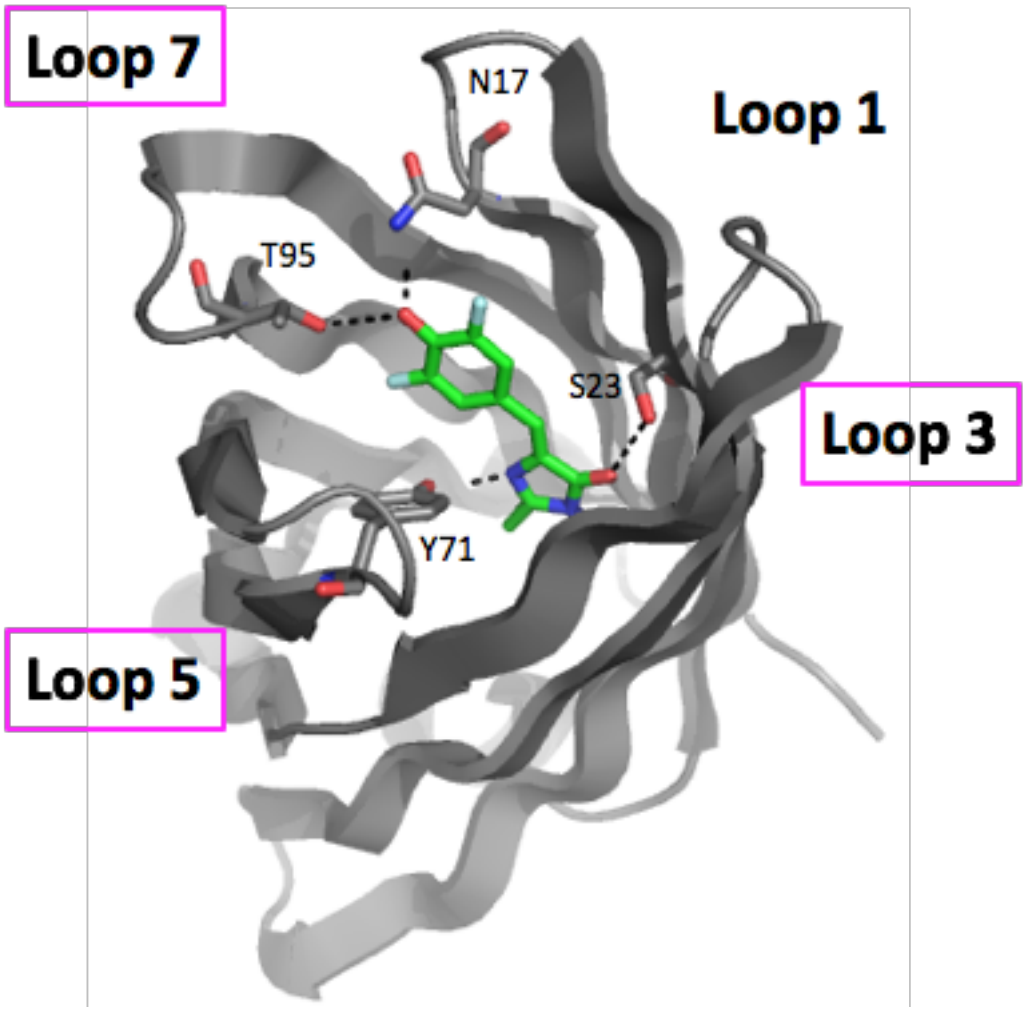

Figure 3: HBI32 designable loop regions and ligand binding residues.

Given the possibly of extensive inter-segment interaction, we focused redesign on four regions within HBI32: loops 1, 3, 5 and 7. For fragment-based library assembly we selected three fragment regions, each spanning two designed loops allowing design of any individual loop or simultaneous design of the two loops on a single fragment.

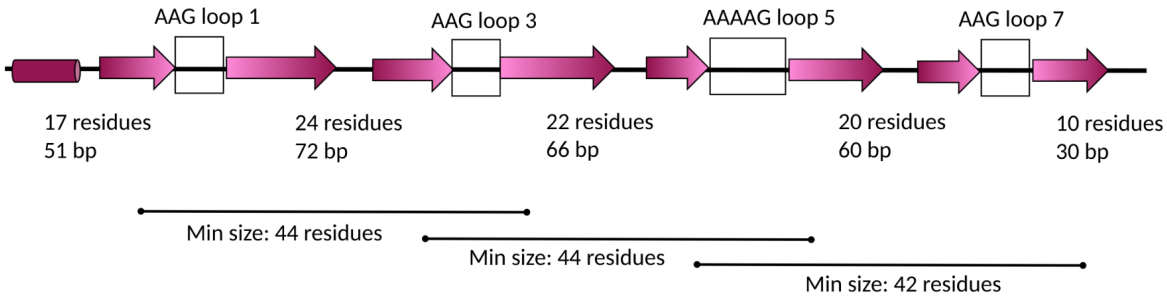

Figure 4: HBI32 fragment-based library construction strategy.

As a starting point for conformation-guided sampling we generated a set of truncation models beginning from HBI32 and a small set of characterized variants generated via previous loop redesign. We generated discontiguous starting models each loop region by excising all non-sheet-identified residues and between 0 and 3 sheet-identified residues extending into the core of the barrel. Critical ligand-binding residues and conserved glycine residues were marked and their sequence identity was preserved in all further design steps.

Conformational samples for each starting model were generated via the conformational sampling pipeline described above. We utilized a 1.0A endpoint lookup tolerance after observing low conformational diversity with the standard .75A lookup tolerance, an 8 residue maximum insertion length and 1.6A inter-segment pruning tolerance. Loop segments were refined via fixed-backbone packing with sequence profile constraints, conserving native proline, glycine and labeled ligand-binding residues, and the top 64 models were selected via hierarchical clustering and unconstrained total score. An equivalent control pool was generated via without sequence profile constraints via an otherwise identical protocol. To assess the effect of fixed backbone design the selected segment conformations were redesigned via a standard FastDesign & LayerDesign based protocol.

A negative control pool was generated by randomly permuting the designed segment sequence, preserving labeled ligand-binding residues. A positive control pool was generated from the native sequences of the characterized starting models.

32 models were selected by total score from single pool and then used as starting points for combinatorial design. Each candidate pair was co-refined via fixed-backbone packing with sequence profile constraints and 128 models in each pool were selected by unconstrained segment total score **and** segment interaction score. To assess the effect of co-design a separate combinatorial library was generated for each loop pair via random selection of designed single segments.

Table 4: Combinatorial Design Benchmark Library Composition

Library Component	Count
Single Segment Redesign with Sequence Profile	5777
Single Segment Redesign without Sequence Profile	6035
Single Segment Resign via FastDesign & LayerDesign	5131
Single Segment Scrambled Negative Control	2500
Single Segment Source Scaffold Controls	64
Segment 1+3 Combinatorial Design	8638
Segment 3+5 Combinatorial Design	11255
Segment 5+7 Combinatorial Design	3290
Segment 1+3 Random Combinatorial Control	2500
Segment 3+5 Random Combinatorial Control	2500
Segment 5+7 Random Combinatorial Control	1000
SV27 Loop Redesign	795

1.3.5 Protein Interface Redesign

To assess the extension of lookup-based turn into functional design, we performed in-context diversification and redesign of a previously characterized PD-1 binder, GR918.3, followed by a high-throughput screening for interface affinity. This design targets the PD-1/PD-L1 interface on murine PD-1, a highly conserved interface between the human and murine variants. As anti-PD1 antibodies and natural ligands are broadly cross-reactive between the two variants we hypothesized that redesign of an existing murine PD1 (mPD-1) may function as a starting point for design of a higher-affinity human PD1 (hPD-1) binder.

We utilized the RosettaScripts based protocol components described above to generated diversified structural samples of GR918.3 modeled in complex with hPD-1. A beta hairpin region adjacent to the binding interface was excised and conformationally resampled via direct segment lookup with loose, .8A, endpoint RMSD tolerance. Designs broadly compatible with the bound conformation, as measured by increased buried interface surface area and centroid-level design-to-target repulsive potential, were selected and designed via Rosetta `FastRelax` with segment sequence profile constraints in complex with hPD-1.

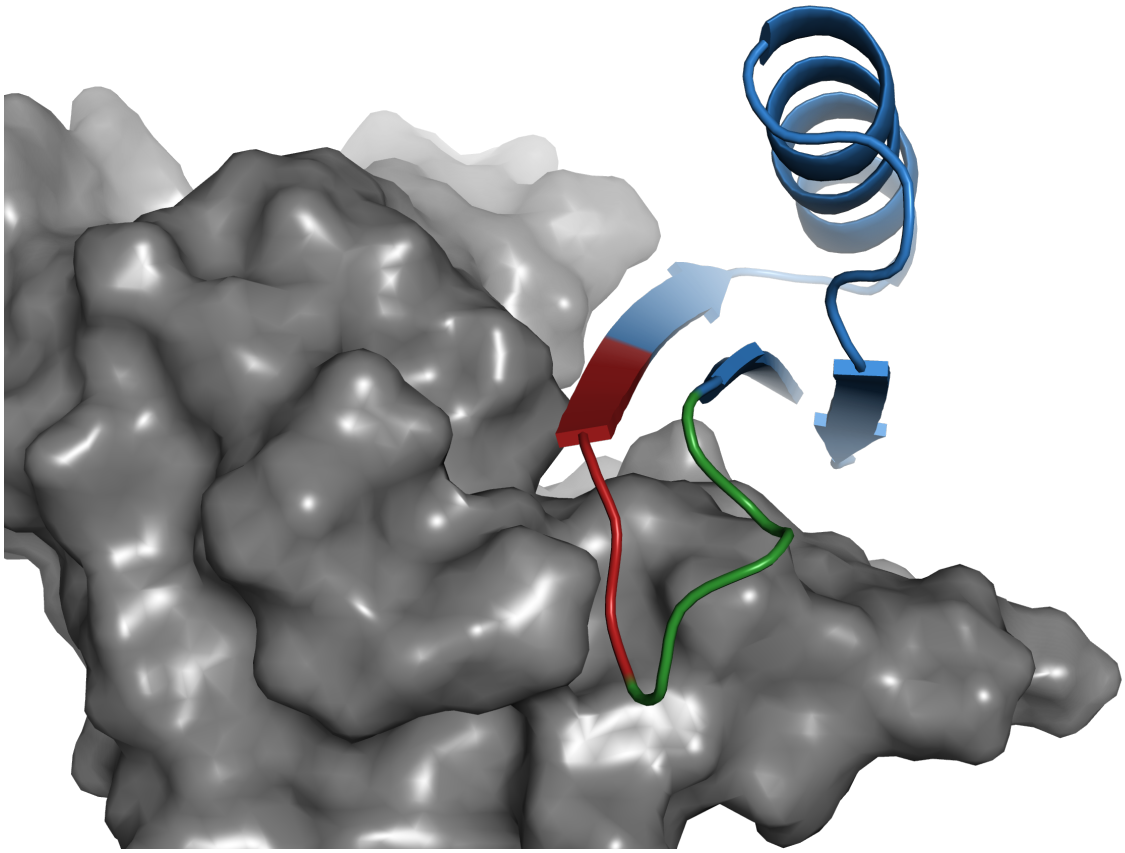

Figure 5: Overview of GR918.3/PD-1 redesign. hPD-1 (grey), GR918.3 scaffold (blue), designed interface region (red), designed loop region (green).

Genes encoding 689 designed variants were ordered and screened for enrichment via Aga-1 yeast surface display against labeled hPD-1 and mPD-1. On screening none of the design variants showed reproducible enrichment against mPD-1. The small number of clones showing enrichment against hPD-1 did not produce labeling when individually screened via the same system.

A single, unanticipated, frame shift mutation was found showing strong enrichment against hPD-1 was identified in the source design pool, involving five mutations in the designed loop segment in the proposed interface region. Subsequent modeling of the mutant via Rosetta `FastRelax`, starting from the proposed design model, generated an interface model in which the mutated residues formed a hot-spot like interaction with the target surface. (*5*) These mutations, which primarily include the insertion of surface-exposed hydrophobic residues, were universally considered disfavorable by the generated segment sequence profile.

Taken together, this results strongly suggest that the designed models interface models are inaccurate, though they do not conclusively indicate that the redesigned region does not adopt the designed

conformation. Designs may **not** bind due to any number of factors, including inaccuracy of the target model in the purposed binding region, inaccuracy of the remodeled segment, effects of the designed region on the conformation of the preserved binding interface and/or disfavorability of the designed interactions to the target. However, given the observation of observed interactions between PD-1 and alternate binding partners in the region targeted by the redesign, it is more likely that the redesigned models are inaccurate.

1.4 Discussion

1.4.1 Technical Overview

Though the data and query model presented here do not represent a dramatic increase in theoretical complexity over previously published work, the implementation of these primitives enables novel use of these methods in protein design. This is most clearly demonstrated in the use of a novel parametric methods for the design of non-local interactions in tandem with the database-guided conformational search and sequence design described in this work. This hybrid design strategy is enabled by technical improvements in the database implementation enabling significantly higher performance that previous fragment-based models as well as increased model flexibility offered by a dynamic indexing approach.

The layered architecture used in this work allows for efficient implementation of conformation-space and sequence-space based queries by explicitly decoupling the database IO layer from an abstract in-memory data store. This allows independent optimization of IO, here via the use of HDF5 for rapid reads of array-oriented numeric data, and query execution, here via the use of the near-optimal QCP alignment kernel. In contrast, previous fragment-based models used within Rosetta coupled both data input and processing, significantly complicating optimization of either task and introducing significant IO overhead to model execution. The order of magnitude or greater increase in model performance in this implementation is critical in allowing *inline* use of fragment-based models in design.

The use of a flexible, normalized data model with on-demand indexing, as opposed to a chunk-oriented data model (*12*) or pre-indexed data store (*14*) is critical in enabling reuse of the data store for multiple query.

This is clearly evident in the length agnostic structure single and structure pair queries, which offer significant increases in performance and flexibility over alternative turn design protocols. Previous approaches, including both fragment and non-fragment based design methods, rely on a fixed pre-specification of the target turn length, and utilize the target length as a core component of the search process. This is a significant obstacle in instances where the target length is unknown or unspecified, as variable length queries may be highly inefficient or even practically impossible when using static, pre-generated indexes.

1.4.2 Choice of Conformation Dependent Distance Metric

Given the relative complexity of multi-stage design protocols, exposing a continuous, tunable distance metric is invaluable in development of protocol subcomponents. Discretized or hash-based tools, in which a static pre-index is generated for a specific sampling application, can provide performance improvements over the search methods presented here, however they strongly incorporate assumptions over model requirements into the hash construction. This severely restricts application specific tuning when applying the model to new contexts. The ability to relax sampling restrictions through a one-dimensional tuning parameter was extremely important in tuning execution efficiency vs exploration in design, as was required when adapting the segment lookup model from conserved turn design to interface structure diversification.

More abstractly, design requirements are often not easily captured as an efficient distance metric, but the chosen distance metric must, as closely as possible, correlate with the downstream design requirements. While the transform hashing tools previously developed were not strongly affected by the potentially surprising properties of the hash formulation, reapplication in design was severely affected by the choice of rotational distance metric.

While design is often compute-limited, optimization and implementation requirements must be considered within the scope of the complete design pipeline. The cost of seconds on computational time per sampling step **may** be overwhelmed by the relative cost of evaluating each sampling result. For instance, though the exhaustive database search system is orders of magnitude slower than a hash-based lookup, as both require subsequent full-atom design this represents an insignificant increase in the overall cost of a design protocol.

1.4.3 Extension Beyond Conserved Turn Design

Previous parametric helical bundle design projects utilized a kinematic closure algorithm accompanied by manual sequence refinement to generate closed models and sequences for non-parametrically described regions of the bundle structure. Though the resulting structure was anomalously stable and had extremely accurate core packing upon verification via xray crystallography, the designed loop segments were extremely inaccurate and caused distinct distortion of the adjacent helical patterns. (*43*) In contrast, the conformational nearest-neighbor model proved effective in identifying designable turn conformations which were effectively recovered in the resulting crystal structures in this project. Taken together, this provides strong evidence that our lookup-based model *can* effectively recover conserved turn structures in design.

In contrast, the PD1 binder diversification application resulted in *no* successful designs. These designs were universally longer, and represent less highly conserved turn conformations. The starting binder

has proven to be robust, and the interface design protocol has been widely applied to optimize protein-protein interfaces, so we can reasonably infer than the designed models are inaccurate despite a lack of direct stability screening or structural data.

Published work on the design of antibody variable domains has demonstrated the importance of maintaining critical native hydrogen bonding and steric interactions during conformational design. Iterative rounds of design and screening demonstrated unconstrained sequence & conformational exploration generated inaccurate design models that were indistinguishable from, or superior to, native antibody models when assessed by score alone. (*22*)

Though the energy model utilized in that work, `score12`, is demonstrably inferior to current optimized variants of the Rosetta full-atom energy model in both design and prediction tasks, similar principles are likely required for successful design of irregular structural features in more diverse contexts.

The sequence-agnostic conformational sampling and sequence biasing tools presented in this work are demonstratively insufficient for the design of longer, diverse loop conformations. At minimum, evaluation of sequence-conformational concordance post sequence design may be required to determine the accuracy of profiled design.

The conformation and sequence models presented in this work also focus exclusively on local conformational and sequence features. Assessment of non-local structural context, in either a sequence independent or sequence dependent fashion, has already proven to be effective in scoring structure prediction tasks. (*36*) Extension of this approach to design via a non-contiguous structure queries would require only a protocol level extension of the current structure database architecture. The current protocols utilize a simple point-specific sequence model to guide design, however non-local sequence features may be critical in design of larger, non-locally encoded structures.

Alternatively, investigation of conserved model features beyond sequence identity and backbone conformation may be highly effective at improving rapid filtering or assessment of design candidates. Identification of conserved structural features such as intra-segment hydrogen bonding patterns or inter-segment non-local interactions could be supported by extending the current data model. Including, for instance, per-residue full-atom energy terms for all source structures would require relatively little additional storage and may provide an extremely robust platform for further lookup-based model development. Alternate features, such as residue solvent accessibility or chemical context, could be supported via additional dynamic indexing operations over the existing data model.

Extension of the data model with existing sequence-space observations may also improve design performance. Previous work has demonstrated that the use of multiple sequence alignments dramatically improves fragment selection in prediction tasks (*45*) and the use of profile-based sequence constraints has proven effective in guiding turn sequence design. Extension of this data model to include homologous sequence data, either in the form of multiple sequence alignments or pre-rendered point-specific homologous sequence distributions may enable simplified application of these approaches to turn

design.

We believe that diverse sampling provided by our high-throughput beta-barrel combinatorial redesign set will provide a fruitful avenue for further investigation and benchmarking of the design tools presented in this work. This dataset will provide an opportunity to assess the efficacy of the individual components of the design pipeline by allowing direct comparison of success rate with and without conformation-based sequence biases and provide benchmark dataset to support development of the extensions outlined above. Moreover, the expansion of high-throughput stability screening datasets (*44*) into more diverse regions of conformational space will provide valuable opportunities for exploration of local structural design.

References

1. J. N. Onuchic, Z. Luthey-Schulten, P. G. Wolynes, Theory of protein folding: The energy landscape perspective. *Annu. Rev. Phys. Chem.* **48**, 545–600 (1997).

2. X. Hu, H. Wang, H. Ke, B. Kuhlman, High-resolution design of a protein loop. *Proc. Natl. Acad. Sci. U. S. A.* **104**, 17668–17673 (2007).

3. N. Koga *et al.*, Principles for designing ideal protein structures. *Nature*. **491**, 222–227 (2012).

4. N. P. King *et al.*, Accurate design of co-assembling multi-component protein nanomaterials. *Nature*. **510**, 103–108 (2014).

5. S. J. Fleishman *et al.*, Hotspot-centric de novo design of protein binders. *J. Mol. Biol.* **413**, 1047–1062 (2011).

6. C. E. Tinberg *et al.*, Computational design of ligand-binding proteins with high affinity and selectivity. *Nature*. **501**, 212–216 (2013).

7. A. Zanghellini *et al.*, New algorithms and an in silico benchmark for computational enzyme design. *Protein Sci.* **15**, 2785–2794 (2006).

8. RefSeq: NCBI reference sequence database.

9. R. P. D. Bank, PDB statistics.

10. R. Bonneau *et al.*, Rosetta in CASP4: Progress in ab initio protein structure prediction. *Proteins*. **Suppl 5**, 119–126 (2001).

11. S. Ovchinnikov, H. Park, D. E. Kim, F. DiMaio, D. Baker, Protein structure prediction using rosetta in CASP12. *Proteins*. **86 Suppl 1**, 113–121 (2018).

12. D. Gront, D. W. Kulp, R. M. Vernon, C. E. M. Strauss, D. Baker, Generalized fragment picking in rosetta:

Design, protocols and applications. *PLoS One*. **6**, e23294 (2011).

13. R. F. Alford *et al.*, The rosetta All-Atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).

14. M. D. Tyka, K. Jung, D. Baker, Efficient sampling of protein conformational space using fast loop building and batch minimization on highly parallel computers. *J. Comput. Chem.* **33**, 2483–2491 (2012).

15. M. D. Tyka *et al.*, Alternate states of proteins revealed by detailed energy landscape mapping. *J. Mol. Biol.* **405**, 607–618 (2011).

16. H. Park *et al.*, Simultaneous optimization of biomolecular energy functions on features from small molecules and macromolecules. *J. Chem. Theory Comput.* **12**, 6201–6212 (2016).

17. S. J. Fleishman, D. Baker, Role of the biomolecular energy gap in protein design, structure, and evolution. *Cell*. **149**, 262–273 (2012).

18. P. Löffler, S. Schmitz, E. Hupfeld, R. Sterner, R. Merkl, Rosetta:MSF: A modular framework for multi-state computational protein design. *PLoS Comput. Biol.* **13**, e1005600 (2017).

19. D. E. Kim, B. Blum, P. Bradley, D. Baker, Sampling bottlenecks in de novo protein structure prediction. *J. Mol. Biol.* **393**, 249–260 (2009).

20. G. D. Lapidoth *et al.*, AbDesign: An algorithm for combinatorial backbone design guided by natural conformations and sequences. *Proteins*. **83**, 1385–1406 (2015).

21. B. Steipe, B. Schiller, A. Plückthun, S. Steinbacher, Sequence statistics reliably predict stabilizing mutations in a protein domain. *J. Mol. Biol.* **240**, 188–192 (1994).

22. D. Baran *et al.*, Principles for computational design of binding antibodies. *Proc. Natl. Acad. Sci. U. S. A.* **114**, 10900–10905 (2017).

23. A. A. Canutescu, R. L. Dunbrack Jr, Cyclic coordinate descent: A robotics algorithm for protein loop closure. *Protein Sci.* **12**, 963–972 (2003).

24. D. J. Mandell, E. A. Coutsias, T. Kortemme, Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling. *Nat. Methods*. **6**, 551–552 (2009).

25. P.-S. Huang *et al.*, RosettaRemodel: A generalized framework for flexible backbone protein design. *PLoS One*. **6**, e24109 (2011).

26. P. Hosseinzadeh *et al.*, Comprehensive computational design of ordered peptide macrocycles. *Science*. **358**, 1461–1466 (2017).

27. G. P. Brady, K. A. Sharp, Entropy in protein folding and in protein-protein interactions. *Curr. Opin.*

Struct. Biol. **7**, 215–221 (1997).

28. V. Geetha, P. J. Munson, Linkers of secondary structures in proteins. *Protein Sci.* **6**, 2538–2547 (1997).

29. W. Li, S. Liang, R. Wang, L. Lai, Y. Han, Exploring the conformational diversity of loops on conserved frameworks. *Protein Eng.* **12**, 1075–1086 (1999).

30. D. E. Engel, W. F. DeGrado, Alpha-alpha linking motifs and interhelical orientations. *Proteins*. **61**, 325–337 (2005).

31. N. C. Fitzkee *et al.*, Are proteins made from a limited parts list? *Trends Biochem. Sci.* **30**, 73–80 (2005).

32. A. Goldenzweig *et al.*, Automated structure- and Sequence-Based design of proteins for high bacterial expression and stability. *Mol. Cell*. **63**, 337–346 (2016).

33. D. L. Theobald, Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr. A.* **61**, 478–480 (2005).

34. Gael Guennebaud and Benoit Jacob and others, Eigen v3.

35. W. Jakob, J. Rhinelander, D. Moldovan, Pybind11 – seamless operability between c++11 and python (2017).

36. J. Zhou, G. Grigoryan, Rapid search for tertiary fragments reveals protein sequence-structure relationships. *Protein Sci.* **24**, 508–524 (2015).

37. Boost geometry module.

38. S. J. Fleishman *et al.*, RosettaScripts: A scripting language interface to the rosetta macromolecular modeling suite. *PLoS One*. **6**, e20161 (2011).

39. Dask Development Team, Dask: Library for dynamic task scheduling (2016).

40. F. Richter, A. Leaver-Fay, S. D. Khare, S. Bjelic, D. Baker, De novo enzyme design using rosetta3. *PLoS One*. **6**, e19230 (2011).

41. C. F. F. Karney, Quaternions in molecular modeling. *J. Mol. Graph. Model.* **25**, 595–604 (2007).

42. S. E. Boyken *et al.*, De novo design of protein homo-oligomers with modular hydrogen-bond network-mediated specificity. *Science*. **352**, 680–687 (2016).

43. P.-S. Huang *et al.*, High thermodynamic stability of parametrically designed helical bundles. *Science*. **346**, 481–485 (2014).

44. G. J. Rocklin *et al.*, Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*. **357**, 168–175 (2017).

45. R. Bonneau, C. E. Strauss, D. Baker, Improving the performance of rosetta using multiple sequence alignment information and global measures of hydrophobic core formation. *Proteins*. **43**, 1–11 (2001).