Designing functional macromolecules

by

Kale Kundert

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Biochemistry and Molecular Biology

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

[redacted]

Think it possible that you may be mistaken.

Oliver Cromwell

Acknowledgements

I would like to gratefully acknowedge the caring support given to me by my wife, Christina Fitzsimmons, throughout my PhD. It's a wonderful luxury having a teammate who's in your corner through thick and thin, and who understands what it's like to be a scientist. I also would like to acknowledge my parents, Mary Glanville and Ken Kundert, who skillfully nurtured and encouraged my interest in science and engineering. Furthermore, I want to thank Tanja Kortemme, my advisor, for giving me the space to explore my own ideas, even those that weren't very good. Finally, I want to thank all the members of the Kortemme lab — Tina Perica, Kyle Barlow, Anum Glasgow, Daniel Hoersch, Ryan Ritterson, Samuel Thompson, Amanda Loshbaugh, XingJie Pan, James Lucas, Cody Krivacic, Chris Mathy, Shane Ó Conchúir, Amelie Stein, Roland Pache, Noah Ollikainen, Laurens Kraal, Yao-ming Huang, and Joslyn Polzien — who made coming to work each day something to look forward to.

Designing functional macromolecules

Kale Kundert

Biology is driven by functional macromolecules, most notably proteins and non-coding RNAs. Learning how to design similarly functional macromolecules is a natural goal. Success will not only bring the ability to create new biological systems, but also the ability to more finely study and manipulate existing biological systems. In this thesis, I will describe two design projects that I pursued over the course of my PhD. The first is a project to remodel the backbone of a protein for the purpose of accurately positioning a catalytic sidechain. The second is a project to ligand-sensitive guide RNAs for the CRISPR-Cas9 system. There is of course much more to be done before we can say that we are able to design functional macromolecules, but the projects described herein move us closer to that goal.

Contents

List of Tables

2 Protein backbone remodeling

3 Ligand-sensitive sgRNAs

List of Figures

2 Protein backbone remodeling

3 Ligand-sensitive sgRNAs

List of Data Files

2 Protein backbone remodeling

Chapter 1

Computational design of structured protein loops

Structured loops are an element of protein structure with special importance for functional proteins. Unlike the canonical elements of protein structure — α-helices and β-sheets — loops can adopt a broad range of conformations because they are not defined by regular geometries or patterns of H-bonds between the polar atoms of the peptide backbone. Similarly, loops can be either rigid or flexible (or rigid in some parts and flexible in others) depending on the interactions they make with themselves and their environments. This conformational and dynamical breadth makes loops well-suited for functionally important tasks like positioning active site residues, forming interfaces, and reacting to signals.

The routine design of functional proteins has been a longstanding goal in the field of protein design. Given the prominent and unique ways in which loops can contribute to function, achieving this goal will inevitably require the ability to rationally design loops. But the same conformational and dynamical breadth that make loops functionally useful also makes them challenging to design: each sequence could adopt a vast number of conformations, each mutation could affect the conformation of every other position in the loop, and each residue could be flexible when it should be rigid, or rigid when it should be flexible.

This perspective will cover the progress that has been made in the field of loop design. I will begin by discussing some examples of functional loops found in nature, to illustrate the applications that loop design aims to enable. I will then continue by reviewing the various efforts that have been made to design loops to date, before concluding by discussing some promising ways for the field to continue moving forward. I believe that the field of loop design is on the verge of significant achievement, and hope that the ideas shared in this perspective can contribute in some way to that achievement.

1.1 Functional loops in nature

Many examples of functional loops can be found in enzymes. In fact, loops are much more common in active sites (50% of residues) than they are in general (30% of residues) [1]. This observation draws attention to the number of ways in which loops can contribute to catalysis. One way is simply by positioning the necessary functional groups. An example of this is the diffusion-limited enzyme ketosteroid isomerase (KSI), in which a catalytic general base (Asp38) positioned by a structured loop isomerizes a double bond. Double mutant cycles have been used to estimate that the positioning provided by the loop has a 1700x effect on k_{cat} [2]. As the loop contains both a cis-proline and a glycine in “right-handed” Ramachandran space, it is unlikely that the same positioning could have been provided by a conventional secondary structure element [2]. Another way that loops can contribute to catalysis is by acting as a lid for the active site. A prototypical example of this is triose phosphate isomerase (TIM). Upon substrate binding, an active site loop moves over 7Å to surround the substrate and hydrogen-bond (H-bond) with the substrate’s phosphate group. This dramatic movement excludes solvent from the reaction and prevents reactive intermediates from escaping the active site [3]. It also limits the rate of product release, highlighting a carefully balanced trade-off between creating an isolated active site and allowing the product to leave. The active site loop is mostly pre-structured, moving only in a hinge region, suggesting that it has been optimized to reduce the entropy penalty of closing [4]. Rationally designing similar systems will require exquisite finesse.

Structured loops also play an important role in protein-protein interactions. Perhaps the most prominent examples of this are antibodies, which use six structured loops — each called a complementarity determining region (CDR) — to bind an astonishing breadth of targets with high affinity and specificity. An examination of these interfaces reveals how loops can contribute to binding.

First is through shape-complementarity. As antibody CDRs mature, they become more complementary to their antigen, which allows for more favorable van der Waals and H-bonding interactions [5, 6]. Second is through pre-organization, which reduces the conformational entropy penalty of antigen binding [7, 8, 9]. However, pre-organization is not a universal feature of high-affinity antibodies [10]. Antibodies with less organized CDRs may benefit from more favorable enthalpic interactions or the ability to bind their antigen in multiple modes [11, 12]. The challenge for rational design will be to create loops that can similarly adopt the surfaces and motions necessary for tight binding.

More examples of functional loops can be found in proteins that react to their environment. One example of this is the bacterial outer membrane protein G (OmpG) which forms a pH-gated pore in the membrane. The gating is mediated by a extracellular loop containing two histidine residues [13, 14]. At basic pH, the histidines are neutral and cohabit adjacent strands of the β-barrel that forms the pore. At acidic pH, the histidines become charged and their strands unzip separate the charge. This results in the loop becoming longer and adopting a conformation which covers the pore. Another prominent example is the activation loop present in protein kinases. When phosphorylated, this loop forms contacts that stabilize the active site and contribute to catalysis. When unphosphorylated, the loop is disordered and catalysis is impaired [15]. These examples illustrate the utility of being able to design and balance multiple functional loop conformations.

1.2 Loop design: The state of the art

In spite of the numerous applications for loop design, there are precious few reports of loops being redesigned. The first such report that I am aware of was an effort to improve a monomeric variant of TIM by restabilizing an 8-residue active site loop that, in wildtype TIM, participated in the dimer interface [16]. The defining feature of this report is that the mutations were chosen manually. In four iterations, computational models of the loop were predicted using Monte Carlo simulations, then mutations were manually proposed to fix various defects in the models. The final result was a 7-residue loop that improved the activity of monomeric TIM. Furthermore, a crystal structure of the designed protein agreed well with the predicted loop conformation well (0.5Å C/Cα/N/O RMSD). This report established very early on that loop design is both achievable and useful.

Another report of manual loop design was made more recently. In this case, players of FoldIt [17] were asked to improve a computationally designed Diels-Alderase [18] by designing an active site loop that would better desolvate the substrate [19]. In the first round of design, the players were allowed to make 5-residue insertions into any of the four active site loops. The authors experimentally tested the 4 best designs (as judged by score and by eye) and over 500 variants. In the second round of design, the players were instructed to stabilize the best first-round design through the creation of a helix-turn-helix motif. This time, the authors tested the 2 best designs and over 400 variants. The result was a variant with a 13-residue insertion that improved catalysis by 150x. A model of the final variant was also created players, and was similar to the crystal structure except for a rotation in one of the helices (3.1Å C/Cα/N/O RMSD). Although the design process required testing hundreds of variants, it clearly demonstrated that human intuition can guide the design of long and functional loops. Ultimately, though, in order for loop design to become a scalable and routine technique, the actual design aspect must be done computationally.

The first report to attempt automated loop design was an effort to graft a loop from an unrelated protein into the fibronectin type III (FN3) domain [20]. This domain had already been established as a non-antibody scaffold for evolving loop-based binding interfaces, and like an antibody, it has a β-sandwich fold that presents 3 mutation-tolerant loops. The aforementioned report redesigned the first of these loops by searching for 12-residue fragments in the protein data bank (PDB) with similar take-off and landing points (within 3Å), grafting each of those loops into the FN3 scaffold, repairing the resulting (small) discontinuities in the backbone, then optimizing the sequence of the new loop while allowing very slight backbone movement (≈0.3Å C/Cα/N/O RMSD, i.e. similar to the average coordinate error in a typical crystal structure). Three designs were purified and two were successfully crystallized. One had the intended loop conformation (0.46Å RMSD), but was in almost the same conformation as the original loop (0.77Å RMSD). The other was missing density for the loop, presumably indicating the lack of defined structure. The significance of this report is that it demonstrated for the first time that a structured loop could be computationally designed. However, this report is also limited: Only 1 of the 3 loops in the scaffold was redesigned, the coordinates of the new loops were taken directly from existing proteins, and the only successful design was in nearly the same conformation as the wildtype loop.

Some of these limitations were addressed by another report in which *de novo* loops were computationally designed in a *de novo* scaffold assembled from 24 repeats of a 5-residue motif [21]. The loops were designed by inserting residues in the middle of the scaffold, sampling them with a coarse-grained and sequence-independent algorithm, then reconstructing them in full-atom detail and performing fixed-backbone sequence optimization. This produced 4000 loop designs. The conformations represented by these designs (which remained sequence-independent) were assumed to approximate the ensemble of states accessible to an 8-residue loop, so the pseudo-probability that each design would fold into its intended conformation was calculated by threading the design sequence onto each design model and comparing the resulting Boltzmann-weighted scores. The 10 designs that were most predicted to fold correctly were tested. Of these, 5 could be purified and 4 could be crystallized. All of the crystal structures were low-resolution (>3.5Å), but two were consistent with their design models, 1 was inconsistent with its model, and 1 was missing density for the loop. This report showed that it's possible to design loops with fully *de novo* conformations, but important limitations remain: the loops were not designed to achieve any particular structure or function, and only a small fraction of the tested designs could be shown to adopt the intended loop conformation.

The effort to change the substrate specificity of human guanine deaminase (hGDA) from guanine to ammelide was the first report of computational loop design being used to achieve a desired function [22]. The ultimate goal was to change the substrate specificity of hGDA from guanine to cytosine, but ammelide was chosen as an intermediate step because it resembles guanine on one face and cytosine of the other. Where hGDA binds guanine using arginine (Arg) and phenylalanine (Phe), it would need either an asparagine (Asn) or glutamine (Gln) to bind ammelide instead. Consequently, the design goal was to remodel the Arg/Phe loop in hGDA to instead position Asn or Gln with the right geometry to bind the cytosine-resembling face of ammelide *. The loop was remodeled by positioning the ends of the Asn and Gln sidechains ideally with respect to ammelide, rotating the sidechain χ angles to generate backbone conformations capable of supporting that ideal positioning, superimposing segments from the scaffold on those backbones, randomly adding or removing residues from either end of those segments, and repairing the backbone with Rosetta. Designs were then made by performing fixed-backbone sequence optimization on the

*Interestingly, in cytosine deaminases the Asn/Glu is be positioned by a different active site loop, so this project is really attempting to build a novel active site architecture.

lowest-scoring backbone model (which featured Asn and two deletions). A single design (GNGV) was chosen for experimental characterization, based on visual inspection and the results of an unrestrained loop modeling simulation. The chosen design effected a 100x increase in ammelide deaminase activity, along with a 25,000x decrease in guanine deaminase activity. A crystal structure revealed that the loop was close to its model (1.0Å Cα RMSD), but that the designed Asn was not pre-organized. This report is significant because it showed that loops can be designed for function, and because the authors remarkably needed to test a only single design (giving hope that loop design can eventually become routine). But there is still clear room for improvement. The designed loop was short and its conformation was only slightly different than wildtype. If we are to employ loops to their full effect, we must learn how to design larger loops and more dramatic conformational changes.

Loop design has also been applied to the very difficult problem of designing antibody CDRs to bind particular targets of interest. This is an especially challenging problem for a number of reasons: (i) there are 6 CDRs, which interact with each other to form a single interface, (ii) some of the CDRs, most notably H3, are very long, and (iii) the position of the antigen is not fixed, and must be optimized in concert with the CDRs. However, there is also an exceptional amount of sequence and structural data available for antibodies, and two groups have reported leveraging this data to rationally design antibody binding interfaces [23, 24]. The first report is based on the idea that each CDR (except H3) can be assigned to a small number of conformational clusters [25]. By combining loops from every possible cluster, 4500 models are created. The epitope is then docked against each model, and the models are designed subject to sequence restraints derived from the natural sequence profiles for each cluster. Each loop is then optimized by iteratively installing different conformations from the same cluster, repacking the sidechains, and minimizing [23]. With the benefit of manual design and directed evolution, this algorithm produced antibodies for two different targets, both with mid-nanomolar affinities. One of these antibodies was crystallized and showed atomic-level accuracy in of 4 of the 6 CDRs (backbone and sidechain), with the only errors being in H1 and the notoriously difficult H3 [26]. The second report is based on mimicking the natural process by which low-affinity germline antibodies undergo mutation and mature into high-affinity binders [24]. The epitope is first placed in various positions relative to the antibody framework, then CDRs from a database are grafted in to create binding interfaces. These interfaces are relaxed in

100 ns molecular dynamics (MD) simulations. If the epitope stays in the binding pocket designed for it, the interface considered analogous to a low-affinity germline antibody and is matured via *in silico* design. This approach produced produced low-nanomolar binders for a dodecapeptide, but the accuracy of the design models cannot be judged since no crystal structure was solved. Together, these methods suggest that it is possible to design large loops, even while also optimizing other degrees of freedom (e.g. epitope docking). They also offer another glimpse of the potential that loops have to provide valuable functionality. The drawback to these methods is that they are dependent upon the vast amount of information available for the antibody scaffold. It is possible that other common scaffolds, e.g. TIM-barrels, might also be amenable to this kind of design, but there remains a need for methods that can be applied to any scaffold.

Having discussed what loop design is, let us briefly discuss some related fields that I consider to be distinct and outside the scope of this review. First is flexible backbone design. While it is well-known that small amounts of backbone motion can dramatically improve sidechain packing [27], this small amount of motion does not seek to move the backbone into a functionally different conformation. In contrast, loop design does seek to move the backbone into functionally different conformations. Second is loop grafting. The goal in loop grafting is to present a fragment of one protein, in its native conformation, on the scaffold of another [refs]. Most often this is done to create an epitope, so that antibodies can be raised against an otherwise recalcitrant antigen. While loop grafting, like loop design, aims to create loops in a particular conformation, it is distinct from loop design because the conformation in question has a known and immutable sequence and structure. This takes the focus off the loop itself and puts it on finding a good scaffold and creating a compatible environment. Third is turn design. An important part of designing *de novo* folds is designing good turns to connect secondary structural elements [28]. This is distinct from loop design because the conformation of the loop doesn't matter so long as it connects the secondary structural elements in question and folds efficiently. Turn design is also a problem that is well-addressed by simple database searches, since small turns have only a limited number of favorable conformations [29].

1.3 What can we learn from loop modeling?

With the current state of rational loop design in mind, it's interesting and worthwhile to consider how the field might progress in the near future. One way to do this is to examine the related — but much more mature — field of loop modeling. Loop modeling is the problem of trying to predict the structure of a loop from its sequence. This is the inverse of the loop design problem, which could be framed as trying to predict sequences that will adopt a particular loop structure. More generally, loop design can be framed as trying to predict sequences that will satisfy certain functional restraints, e.g. positioning one or more sidechains, adopting a particular conformation, changing conformation in the presence of a ligand, etc. By carefully considering the similarities and differences between these two related problems, we will see how previous advances in loop modeling can illuminate the way forward in loop design.

The basic structure of a loop modeling algorithm is as follows: The inputs are (i) the sequences of one or more loops and (ii) the atomic coordinates for the rest of the protein. For example, these coordinates might come from homology models or experimental structures with missing atoms. The outputs are the atomic coordinates for loops in question. To produce these coordinates, a loop modeling algorithm needs four components: a way to represent the atoms in question, a way to sample new loop conformations, a way to keep the backbone closed, and a way to score different loop conformations. I will discuss each of these components, and how they might be applied to the loop design problem, below.

1.3.1 Representation

Almost every loop modeling algorithm makes use of two representations: one that's coarse-grained and another that's full-atom. A coarse-grained representation is one that strips away some atomic detail in the interest of simplicity. This could mean replacing the sidechain atoms with a single large sphere, or removing the sidechain atoms altogether, or removing everything except the α-carbons. In contrast, a full atom representation includes every backbone and sidechain atom, although most still exclude solvent atoms. The advantage of coarse-grained representations is that they create smaller and smoother energy landscapes which can be thoroughly explored, while the advantage of full-atom representations is that they allow for important physical interactions, like hydrophobic packing and H-bonding, to be modeled. For this reason, most loop modeling methods begin by searching for reasonable loop conformations in a coarse-grained representation, then switch to a full-atom representation to winnow and refine those conformations [30, 31, 32, 33, 34, 35]. An interesting exception is an algorithm that uses only a full-atom representation [36, 37]. It is based on the premise that the best loop conformation will comprise the best residue conformations, so it build loops by sampling each residue in full-atom detail, one-at-a-time, until the whole loop has been assembled.

The clear consensus from the loop modeling literature is that it's best to use both coarse-grained and full-atom representations. However, the variety of coarse-grained representations that can be used for loop design is limited by the need to represent sequence. Loop design is fundamentally a search for sequences, so in order to perform a coarse-grained version of this search, the representation must encode sequence. Even defining the objective of a loop design effort can depend on the sidechains being represented. For example, if the goal is to position the functional group of an active site residue, solutions will need to take into account the size and geometry of that residue's sidechain, even at a coarse-grained level. In short, coarse-grained representations that ignore the sidechains altogether will be less appropriate for loop design. The coarse-grained representation in Rosetta (termed "centroid-mode") may be a good candidate moving forward, as it represents different sidechains as spheres with different sizes and polar properties [38]. It may also be worthwhile to develop new representations specifically for the loop design problem.

1.3.2 Sampling

The easiest way to distinguish two loop modeling algorithms is by how they sample different conformations. Algorithms are traditionally categorized as either "template-based" or "template-free" [39, 40, 41], where the former query databases of known structures to sample loop conformations, and the latter don't. However, most algorithms lie on a continuum between the two. On one side of this continuum are the algorithms that make no direct use of structural data. One way to do this is to randomly place atoms and subsequently refine them to satisfy certain physical or experimental restraints [30, 42, 43]. Another way is to make small perturbations to the backbone coordinates, in either Monte Carlo [44, 45] or molecular dynamics (MD) [46, 47, 48, 49] simulations. The first step along the continuum is to sample backbone torsions from the Ramachandran distribution, which is derived from the frequencies of different combinations of the φ and ψ backbone torsions in high-resolution protein structures. This is a perhaps the most popular strategy [32, 34, 50, 51, 52, 53, 54, 55, 56], and has even been extended to two-residue [57] and three-residue [58] versions of the Ramachandran distribution. Next are the algorithms that sample new loop conformations by stitching together fragments (usually of about 3–9 residues) from known structures [33, 35, 59]. This fragment-based approach posits that all relevant local conformations are present in the PDB, and is widely recognized for its successful application to the *ab initio* prediction of protein tertiary structures [60]. Finally, on the far side of the continuum are the fully template-based algorithms. These algorithms query structural databases for loops of the right length that roughly match the takeoff and landing points of the loop in the input structure [61, 62, 63, 64, 65, 66, 67, 68, 69]. Matching loops are usually ranked by how well they fit the gap and align with the input sequence, and can be subsequently relaxed using a full-atom score function.

In terms of sampling, the clearest difference loop modeling and design is that the former only needs to sample conformation-space, while the latter needs to simultaneously sample conformation- and sequence-space. I will put aside the issue of sampling sequence-space, as it is not informed by the aforementioned literature, and focus instead on the issue of sampling conformation-space.

Due to the lack of fair and comprehensive benchmarks between loop modeling methods [40], I can't judge which has been the most successful for loop modeling. However, I can speculate that the template- and fragment-based algorithms will be the most successful for loop design [22, 23, 70]. The reason is that these algorithms offer a solution to the "designability" problem [71]: Given a desired conformation, is it possible for some sequence (in some environmental context) to adopt that conformation? If the desired conformation came from a structural database, the answer is yes. There are two challenges in applying purely template-based algorithms to the problem of loop design. The first is ensuring that the loop will still adopt the desired conformation in its new context. The second is that there will be extra geometrical constraints on the loop. For example, to design a loop that positions an active site residue, a database query would have to find loops that not only start and stop in the right place, but also are capable of positioning the residue in question. This challenge only gets worse as more residues are included in the design. For example, an interface design project might require that *every* residue in the loop contributes to binding! That said, loop design also makes the database query easier in other ways — the algorithm can pick its takeoff and landing points, and the loop can be of any length or sequence — so it's not clear *a priori* how difficult it will be to apply template-based algorithms this new problem. Either way, the fragment-based algorithms are another promising choice. They offer similar advantages to the template-based algorithms in terms of designability, but can also easily accommodate restraints imposed by the design goal, e.g. with extra score terms.

Another aspect of sampling is the question of how large barriers are traversed. The most common answer to this question is simulated annealing, whereby the temperature of the simulation is gradually increased and decreased over the course of the simulation [30, 33, 34, 44, 45, 46, 54, 55, 59]. A closely related alternative is parallel tempering, whereby simulations at different temperatures are run simultaneously and occasionally swap coordinates [48, 72]. The advantage of this technique is that it produces ensembles with defined temperatures, but the proper treatment of thermodynamic ensembles has not been a priority for the field. Genetic algorithms have also been used to enhance sampling [43, 73, 74]. While genetic algorithms can traverse barriers very efficiently, they also have to confront the fact that crossover operations involving backbone torsions are likely to produce large clashes[75]. Lastly, a handful of methods have attempted to exhaustively sample conformational space, subject to some binning [32, 36, 37, 53].

Any of these barrier traversal strategies could be effectively applied to the part of a loop design protocol that involves searching for sequences and conformations that satisfy the design criteria. But once that that part of the protocol produces some candidate sequences, the next part needs to assess which sequences will really adopt the intended conformation. This validation step is similar to a loop modeling simulation, but it's simplified in one way: It's testing the hypothesis that the intended conformation is the global energy minimum, so it can stop as soon as it finds evidence to the contrary. If a small number of plausible off-target states could be identified (or perhaps even recalled from the simulations that produced the candidate in question), the validation problem could be recast as a comparison between those states, rather than as a global search for the energy minimum. In turn, this may justify the use of enhanced sampling techniques like umbrella sampling [76] or the adaptive biasing force method [77].

1.3.3 Closure

A unique feature of loop modeling algorithms is that they must sample new loop conformations without creating breaks in the protein backbone. This is referred to as the closure problem. The simplest solution is to simply start building the loop from both ends, and to keep models that happen to meet in the middle [32, 36, 52]. This is a common approach for sampling algorithms that are enumerative in some way. Another solution is to define some kind of score term that favors a closed backbone (e.g. a harmonic restraint across the break) and to let the sampling algorithm (or a gradient minimizer) find ways to satisfy that term [30, 42, 43, 44, 45, 53, 54, 56, 59, 78]. However, this solution may require spending a significant amount of time sampling conformations that aren't even closed, which is inefficient. An alternative is to use inverse kinematics algorithms borrowed from the field of robotics. These algorithms seek to solve the following problem: If you have a robot arm with multiple joints, and you want the end of the arm at some given position and orientation, to what angle should you set each joint? In the context of loop modeling, such algorithms can be used after sampling to adjust the backbone torsions in the loop such that its ends remain connected to the rest of the protein. There are many inverse kinematics algorithms, but they can be broadly categorized as either iterative or analytical. Iterative algorithms converge on a closed backbone over a series of steps, as exemplified by cyclic coordinate descent (CCD) [79]. These algorithms are conceptually simple and have been applied in many protocols [33, 51, 55, 68, 73, 80, 81]. Analytical algorithms calculate exact solutions to the closure problem, as exemplified by kinematic closure (KIC) [82]. Since the end of the "robot arm" has 6 degrees-of-freedom (3 positional and 3 orientational), these algorithms must set 6 backbone torsions to achieve closure. Any other torsions are unaffected, no matter how long the loop is. These algorithms are more complicated, but have the nice properties of perturbing the minimum number of torsions and indicating immediately if closure is possible. They have also been used in many protocols [34, 35, 37, 74, 83].

Loop design will require efficient sampling in sequence- and conformation-space. For this reason I believe that the efficiency of the inverse kinematics methods, especially the analytical ones, will make them the best choices for maintaining closed loops.

1.3.4 Scoring

The last component of a loop modeling algorithm is the score function used to evaluate which conformations are the most realistic. As with sampling, loop modeling algorithms lie on a continuum based on the score function they employ and how much structural data it makes use of. On one side of the continuum are the algorithms that use physical score functions like AMBER [46, 49], CHARMM [48, 53, 72], and OPLS [32]. Some algorithms also use a "colony" score term that tries to capture the idea of entropy by favoring the models with the most conformationally similar neighbors [51, 84]. These score functions attempt to apply our understanding of physics, in a simplified way, to discriminate between loop models. On the other side of the continuum are algorithms that use statistical score functions like DFIRE [35, 37, 66, 85], DOPE [54], SOAP-Loop [68], and others [45, 50, 86]. These score functions attempt to create rules from the distributions of atoms and residues observed in high-resolution structures, and can be good at implicitly capturing complex effects like secondary structure, sidechain-sidechain interactions (e.g. salt-bridging, π-stacking), and packing defects. However, by far the greatest share of loop modeling algorithms fall in the middle of the continuum and use hybrid score functions, or score functions which include both physical and statistical terms [30, 31, 33, 34, 43, 55, 59, 73, 74]. Hybrid score functions typically include a complete set of physical terms, plus statistical terms that favor common backbone torsions, sidechain torsion, and H-bonding geometries. It's also common for methods to use a statistical score function for coarse-grained sampling and a physical score function for the full-atom sampling.

What considerations are relevant to loop design? Hybrid score functions are a clear consensus, especially among the most recent methods, so I expect their use to continue. A more significant consideration is the need for a score term that allows the for the fair evaluation of mutations. For example, imagine that you were to attempt to mutate an alanine to an arginine. Absent any correction, arginine would artificially score better than alanine simply because it has more atoms, and thus more opportunities to make favorable contacts. A score term is needed to counteract this bias. While in principle such a term could be added to any of the score functions used for loop modeling, the Rosetta score function is the only one of those that already has one (called the "reference energy"). This means that for now, the Rosetta score function is the best candidate for applications in loop design. In keeping with this idea, four of the five computational loop design methods reviewed above used the Rosetta score function (the other ignored this consideration).

Another consideration for loop design is the solvent model. While every loop modeling method that I am aware of uses an implicit solvent model (or doesn't consider the solvent at all), it may be possible to apply explicit solvent models in the context of loop design. As mentioned in the paragraphs on barrier traversal, the validation step of a loop design protocol may be able to devote more time towards a small number of structures, allowing the use of more resource-intensive techniques. As loops are typically solvent exposed, an explicit treatment of the solvent may yield worthwhile improvements in accuracy.

1.4 What problems are unique to loop design?

Having discussed loop design in the context of loop modeling, let us now focus on some problems that are specific to the loop design problem. The first of these is: how many residues should be in the designed loop? It must be long enough to fulfill the design goal (e.g. if the goal is to position a residue, to loop must be able to reach that residue), but ideally as short as possible. Not only are shorter loops less likely to be conformationally heterogeneous, but they are also easier to accurately model. The most naive approach to designing loop length is to simply try several different lengths, but this is inefficient. Loop design already has to grapple with the enormous task of sampling both sequence- and conformation-space. It would be wasteful to sample unnecessary loop lengths on top of that, especially for problems where the loop length may not be well constrained. Two more thoughtful approaches have already been explored. Murphy et al. randomly added and removed residues from the loop during design, and validated their approach with a loop length recovery benchmark [22]. Lapidoth et al. sampled loop sequences and conformations from a database, which included loops of different lengths [23]. However, neither of these approaches used a score function that was capable of fairly comparing loops of different lengths. Just as score functions will naturally prefer large amino acids over short ones (as described above), so too will they prefer long loops over short ones. While this bias did not prevent either group from creating successful designs, it is a shortcoming that should be addressed as the field matures.

The second problem that loop design must confront is: how can the rigidity of a loop be designed? Although it is well known that proteins are best thought of as occupying an ensemble of states at equilibrium, only a handful of loop modeling methods have tried to account for the possibility that a loop might not have a single defined conformation [87, 88, 89]. This may be a niche consideration for loop modeling, where the sequences being predicted have been optimized by evolution and are often well-structured, but it is of immediate importance to loop design, where the sequences being predicted were created *in silico* and disorder could be a common mode of failure. Making a loop more flexible or rigid may also be a design goal. Predicting protein flexibility is an established field, although to my knowledge it had never been applied to loop design. There are two basic approaches. The first is to generate an ensemble of possible conformations, then to calculate Boltzmann-averaged quantities (like RMSD) over that ensemble [87, 88, 90, 91]. The challenge with this approach is the expense of computing the ensembles and the impossibility of knowing whether all of the relevant states have been sampled. The ensembles must also be generated by a method that obeys detailed balance, which adds complexity. The second approach is to represent the protein as a graph and to infer rigidity from the connectivity of that graph [92, 93, 94, 95, 96]. Usually the nodes represent atoms or residues, and the edges represent the covalent and non-covalent interactions between those nodes. The challenge with this approach is that it abstracts the details of protein structure and is often more focused on motions at the domain level than at the individual residue level. It is still an open question which of these two approaches will work best for loop design.

1.5 Closing remarks

In conclusion, I have reviewed the current state of the loop design field and highlighted several promising avenues for progress in the near future. The field has had success designing small loops and antibodies, and can continue making progress by repurposing existing loop modeling algorithms. Questions like how long to make a loop, and how to make a loop either rigid or flexible, still need to be grappled with. That said, I believe that the technologies enabling the next steps forward are largely in place. My hope is these steps will lead to methods capable of routinely and accurately designing structured loops. As loops are an integral feature of many functional proteins — including enzymes, binders, and switches — such methods will be a boon to the broader and ongoing effort to design functional proteins.

1.6 References

[1] Gail J. Bartlett et al. “Analysis of Catalytic Residues in Enzyme Active Sites”. In: *Journal of Molecular Biology* 324.1 (Nov. 2002), pp. 105–121. ISSN: 0022-2836. DOI: 10.1016/S0022-2836(02)01036-7. URL: http://www.sciencedirect.com/science/article/pii/S0022283602010367 (visited on 04/06/2018) (cit. on p. 2).

[2] Jason P. Schwans et al. “Experimental and Computational Mutagenesis To Investigate the Positioning of a General Base within an Enzyme Active Site”. In: *Biochemistry* 53.15 (Apr. 2014), pp. 2541–2555. ISSN: 0006-2960. DOI: 10.1021/bi401671t. URL: https://doi.org/10.1021/bi401671t (visited on 04/06/2018) (cit. on p. 2).

[3] David L. Pompliano, Anusch Peyman, and Jeremy R. Knowles. “Stabilization of a reaction intermediate as a catalytic device: definition of the functional role of the flexible loop in triosephosphate isomerase”. In: *Biochemistry* 29.13 (Apr. 1990), pp. 3186–3194. ISSN: 0006-2960. DOI: 10.1021/bi00465a005. URL: https://doi.org/10.1021/bi00465a005 (visited on 05/14/2018) (cit. on p. 2).

[4] Elias Lolis and Gregory A. Petsko. “Crystallographic analysis of the complex between triosephosphate isomerase and 2-phosphoglycolate at 2.5-.ANG. resolution: implications for catalysis”. In: *Biochemistry* 29.28 (July 1990), pp. 6619–6625. ISSN: 0006-2960. DOI: 10.1021/bi00480a010. URL: https://doi.org/10.1021/bi00480a010 (visited on 05/14/2018) (cit. on p. 2).

[5] Yili Li et al. “X-ray snapshots of the maturation of an antibody response to a protein antigen”. en. In: *Nature Structural & Molecular Biology* 10.6 (June 2003), pp. 482–488. ISSN: 1545-9985. DOI: 10.1038/nsb930. URL: https://www.nature.com/articles/nsb930 (visited on 05/13/2018) (cit. on p. 3).

[6] Daisuke Kuroda and Jeffrey J. Gray. “Shape complementarity and hydrogen bond preferences in protein–protein interfaces: implications for antibody modeling and protein–protein docking”. en. In: *Bioinformatics* 32.16 (Aug. 2016), pp. 2451–2456. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw197. URL: https://academic.oup.com/bioinformatics/article/32/16/2451/2288376 (visited on 05/13/2018) (cit. on p. 3).

[7] Ian F. Thorpe and Charles L. Brooks. “Molecular evolution of affinity and flexibility in the immune system”. en. In: *Proceedings of the National Academy of Sciences* 104.21 (May 2007), pp. 8821–8826. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0610064104. URL: http://www.pnas.org/content/104/21/8821 (visited on 05/13/2018) (cit. on p. 3).

[8] Sergio E. Wong, Ben D. Sellers, and Matthew P. Jacobson. “Effects of somatic mutations on CDR loop flexibility during affinity maturation”. en. In: *Proteins: Structure, Function, and Bioinformatics* 79.3 (Mar. 2011), pp. 821–829. ISSN: 1097-0134. DOI: 10.1002/prot.22920. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.22920 (visited on 05/13/2018) (cit. on p. 3).

[9] Thaddeus M. Davenport et al. “Somatic Hypermutation-Induced Changes in the Structure and Dynamics of HIV-1 Broadly Neutralizing Antibodies”. In: *Structure* 24.8 (Aug. 2016), pp. 1346–1357. ISSN: 0969-2126. DOI: 10.1016/j.str.2016.06.012. URL: http://www.sciencedirect.com/science/article/pii/S0969212616301393 (visited on 05/13/2018) (cit. on p. 3).

[10] Jeliazko R. Jeliazkov et al. “Repertoire Analysis of Antibody CDR-H3 Loops Suggests Affinity Maturation Does Not Typically Result in Rigidification”. English. In: *Frontiers in Immunology* 9 (2018). ISSN: 1664-3224. DOI: 10.3389/fimmu.2018.00413. URL: https://www.frontiersin.org/articles/10.3389/fimmu.2018.00413/full (visited on 04/12/2018) (cit. on p. 3).

[11] Leo C. James, Pietro Roversi, and Dan S. Tawfik. “Antibody Multispecificity Mediated by Conformational Diversity”. en. In: *Science* 299.5611 (Feb. 2003), pp. 1362–1367. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1079731. URL: http://science.sciencemag.org/content/299/5611/1362 (visited on 05/13/2018) (cit. on p. 3).

[12] Wei Wang et al. “Conformational Selection and Induced Fit in Specific Antibody and Antigen Recognition: SPE7 as a Case Study”. In: *The Journal of Physical Chemistry B* 117.17 (May 2013), pp. 4912–4923. ISSN: 1520-6106. DOI: 10.1021/jp4010967. URL: https://doi.org/10.1021/jp4010967 (visited on 05/13/2018) (cit. on p. 3).

[13] Özkan Yildiz et al. “Structure of the monomeric outer-membrane porin OmpG in the open and closed conformation”. en. In: *The EMBO Journal* 25.15 (Aug. 2006), pp. 3702–3713. ISSN: 0261-4189, 1460-2075. DOI: 10.1038/sj.emboj.7601237. URL: http://emboj.embopress.org/content/25/15/3702 (visited on 05/01/2018) (cit. on p. 3).

[14] Tiandi Zhuang et al. “NMR-Based Conformational Ensembles Explain pH-Gated Opening and Closing of OmpG Channel”. In: *Journal of the American Chemical Society* 135.40 (Oct. 2013), pp. 15101–15113. ISSN: 0002-7863. DOI: 10.1021/ja408206e. URL: https://doi.org/10.1021/ja408206e (visited on 05/13/2018) (cit. on p. 3).

[15] Jon M. Steichen et al. “Structural Basis for the Regulation of Protein Kinase A by Activation Loop Phosphorylation”. en. In: *Journal of Biological Chemistry* 287.18 (Apr. 2012), pp. 14672–14680. ISSN: 0021-9258, 1083-351X. DOI: 10.1074/jbc.M111.335091. URL: http://www.jbc.org/content/287/18/14672 (visited on 05/13/2018) (cit. on p. 3).

[16] N. Thanki et al. “Protein engineering with monomeric triosephosphate isomerase (monoTIM): the modelling and structure verification of a seven-residue loop”. eng. In: *Protein Engineering* 10.2 (Feb. 1997), pp. 159–167. ISSN: 0269-2139 (cit. on p. 3).

[17] Seth Cooper et al. “Predicting protein structures with a multiplayer online game”. en. In: *Nature* 466.7307 (Aug. 2010), pp. 756–760. ISSN: 1476-4687. DOI: 10.1038/nature09304. URL: https://www.nature.com/articles/nature09304 (visited on 05/14/2018) (cit. on p. 4).

[18] Justin B. Siegel et al. “Computational Design of an Enzyme Catalyst for a Stereoselective Bimolecular Diels-Alder Reaction”. en. In: *Science* 329.5989 (July 2010), pp. 309–313. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1190239. URL: http://science.sciencemag.org/content/329/5989/309 (visited on 05/14/2018) (cit. on p. 4).

[19] Christopher B. Eiben et al. “Increased Diels-Alderase activity through backbone remodeling guided by Foldit players”. en. In: *Nature Biotechnology* 30.2 (Feb. 2012), pp. 190–192. ISSN: 1546-1696. DOI: 10.1038/nbt.2109. URL: https://www.nature.com/articles/nbt.2109 (visited on 05/13/2018) (cit. on p. 4).

[20] Xiaozhen Hu et al. “High-resolution design of a protein loop”. en. In: *Proceedings of the National Academy of Sciences* 104.45 (Nov. 2007), pp. 17668–17673. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0707977104. URL: http://www.pnas.org/content/104/45/17668 (visited on 11/16/2017) (cit. on p. 4).

[21] James T. MacDonald et al. “Synthetic beta-solenoid proteins with the fragment-free computational design of a beta-hairpin extension”. en. In: *Proceedings of the National Academy of Sciences* 113.37 (Sept. 2016), pp. 10346–10351. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1525308113. URL: http://www.pnas.org/content/113/37/10346 (visited on 04/05/2018) (cit. on p. 5).

[22] Paul M. Murphy et al. “Alteration of enzyme specificity by computational loop remodeling and design”. en. In: *Proceedings of the National Academy of Sciences* 106.23 (June 2009), pp. 9215–9220. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0811070106. URL: http://www.pnas.org/content/106/23/9215 (visited on 11/17/2017) (cit. on pp. 5, 10, 14).

[23] Gideon D. Lapidoth et al. “AbDesign: An algorithm for combinatorial backbone design guided by natural conformations and sequences”. In: *Proteins: Structure, Function, and Bioinformatics* 83.8 (July 2015), pp. 1385–1406. ISSN: 0887-3585. DOI: 10.1002/prot.24779. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/prot.24779 (visited on 03/30/2018) (cit. on pp. 6, 10, 14).

[24] Poosarla Venkata Giridhar et al. “Computational de novo design of antibodies binding to a peptide with high affinity”. In: *Biotechnology and Bioengineering* 114.6 (Jan. 2017), pp. 1331–1342. ISSN: 0006-3592. DOI: 10.1002/bit.26244. URL: https://onlinelibrary.wiley.com/doi/full/10.1002/bit.26244 (visited on 04/06/2018) (cit. on p. 6).

[25] Cyrus Chothia and Arthur M. Lesk. “Canonical structures for the hypervariable regions of immunoglobulins”. In: *Journal of Molecular Biology* 196.4 (Aug. 1987), pp. 901–917. ISSN: 0022-2836. DOI: 10.1016/0022-2836(87)90412-8. URL: http://www.sciencedirect.com/science/article/pii/0022283687904128 (visited on 04/12/2018) (cit. on p. 6).

[26] Dror Baran et al. “Principles for computational design of binding antibodies”. en. In: *Proceedings of the National Academy of Sciences* 114.41 (Oct. 2017), pp. 10900–10905. ISSN:

0027-8424, 1091-6490. DOI: 10.1073/pnas.1707171114. URL: http://www.pnas.org/content/114/41/10900 (visited on 04/06/2018) (cit. on p. 6).

[27] Daniel J Mandell and Tanja Kortemme. “Backbone flexibility in computational protein design”. In: *Current Opinion in Biotechnology*. Protein technologies / Systems and synthetic biology 20.4 (Aug. 2009), pp. 420–428. ISSN: 0958-1669. DOI: 10.1016/j.copbio.2009.07.006. URL: http://www.sciencedirect.com/science/article/pii/S0958166909000913 (visited on 05/11/2018) (cit. on p. 7).

[28] Po-Ssu Huang et al. “De novo design of a four-fold symmetric TIM-barrel protein with atomic-level accuracy”. In: *Nature chemical biology* 12.1 (Jan. 2016), pp. 29–34. ISSN: 1552-4450. DOI: 10.1038/nchembio.1966. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4684731/ (visited on 05/13/2018) (cit. on p. 7).

[29] Peicheng Du, Michael Andrec, and Ronald M. Levy. “Have we seen all structures corresponding to short protein fragments in the Protein Data Bank? An update”. In: *Protein Engineering, Design and Selection* 16.6 (June 2003), pp. 407–414. ISSN: 1741-0126. DOI: 10.1093/protein/gzg052. URL: https://academic.oup.com/peds/article/16/6/407/1511932 (visited on 11/16/2017) (cit. on p. 7).

[30] A. Fiser, R. K. Do, and A. Sali. “Modeling of loops in protein structures.” In: *Protein Science : A Publication of the Protein Society* 9.9 (Sept. 2000), pp. 1753–1773. ISSN: 0961-8368. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2144714/ (visited on 05/08/2018) (cit. on pp. 9, 11–13).

[31] Paul I. W. de Bakker et al. “Ab initio construction of polypeptide fragments: Accuracy of loop decoy discrimination by an all-atom statistical potential and the AMBER force field with the Generalized Born solvation model”. en. In: *Proteins: Structure, Function, and Bioinformatics* 51.1 (Apr. 2003), pp. 21–40. ISSN: 1097-0134. DOI: 10.1002/prot.10235. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.10235/abstract (visited on 11/16/2017) (cit. on pp. 9, 13).

[32] Matthew P. Jacobson et al. “A hierarchical approach to all-atom protein loop prediction”. en. In: *Proteins: Structure, Function, and Bioinformatics* 55.2 (May 2004), pp. 351–367. ISSN:

1097-0134. DOI: 10.1002/prot.10613. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.10613/abstract (visited on 11/16/2017) (cit. on pp. 9–13).

[33] Chu Wang, Philip Bradley, and David Baker. “Protein–Protein Docking with Backbone Flexibility”. In: *Journal of Molecular Biology* 373.2 (Oct. 2007), pp. 503–519. ISSN: 0022-2836. DOI: 10.1016/j.jmb.2007.07.050. URL: http://www.sciencedirect.com/science/article/pii/S0022283607010030 (visited on 05/10/2018) (cit. on pp. 9–13).

[34] Daniel J. Mandell, Evangelos A. Coutsias, and Tanja Kortemme. “Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling”. en. In: *Nature Methods* 6.8 (Aug. 2009), pp. 551–552. ISSN: 1548-7105. DOI: 10.1038/nmeth0809-551. URL: https://www.nature.com/articles/nmeth0809-551 (visited on 05/08/2018) (cit. on pp. 9–13).

[35] Julian Lee et al. “Protein loop modeling by using fragment assembly and analytical loop closure”. en. In: *Proteins: Structure, Function, and Bioinformatics* 78.16 (Dec. 2010), pp. 3428–3436. ISSN: 1097-0134. DOI: 10.1002/prot.22849. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.22849/abstract (visited on 11/16/2017) (cit. on pp. 9, 10, 12, 13).

[36] Rhiju Das. “Atomic-Accuracy Prediction of Protein Loop Structures through an RNA-Inspired Ansatz”. In: *PLOS ONE* 8.10 (Oct. 2013), e74830. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0074830. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0074830 (visited on 11/17/2017) (cit. on pp. 9, 11, 12).

[37] Samuel W. K. Wong, Jun S. Liu, and S. C. Kou. “Fast de novo discovery of low-energy protein loop conformations”. en. In: *Proteins: Structure, Function, and Bioinformatics* 85.8 (Aug. 2017), pp. 1402–1412. ISSN: 1097-0134. DOI: 10.1002/prot.25300. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25300 (visited on 05/11/2018) (cit. on pp. 9, 11–13).

[38] Jeffrey J. Gray et al. “Protein–Protein Docking with Simultaneous Optimization of Rigid-body Displacement and Side-chain Conformations”. In: *Journal of Molecular Biology* 331.1 (Aug. 2003), pp. 281–299. ISSN: 0022-2836. DOI: 10.1016/S0022-2836(03)00670-3. URL:

http://www.sciencedirect.com/science/article/pii/S0022283603006703 (visited on 05/11/2018) (cit. on p. 9).

[39] Amarda Shehu and Lydia E. Kavraki. “Modeling Structures and Motions of Loops in Protein Molecules”. en. In: *Entropy* 14.2 (Feb. 2012), pp. 252–290. DOI: 10.3390/e14020252. URL: http://www.mdpi.com/1099-4300/14/2/252 (visited on 05/08/2018) (cit. on p. 9).

[40] Yaohang Li. “Conformational sampling in template-free protein loop structure modeling: An overview”. In: *Computational and Structural Biotechnology Journal* 5.6 (Feb. 2013), e201302003. ISSN: 2001-0370. DOI: 10.5936/csbj.201302003. URL: http://www.sciencedirect.com/science/article/pii/S2001037014600313 (visited on 11/17/2017) (cit. on pp. 9, 10).

[41] András Fiser. “Comparative Protein Structure Modelling”. en. In: *From Protein Structure to Function with Bioinformatics*. Springer, Dordrecht, 2017, pp. 91–134. ISBN: 978-94-024-1067-9 978-94-024-1069-3. DOI: 10.1007/978-94-024-1069-3_4. URL: https://link.springer.com/chapter/10.1007/978-94-024-1069-3_4 (visited on 11/16/2017) (cit. on p. 9).

[42] Pu Liu et al. “A Self-Organizing Algorithm for Modeling Protein Loops”. In: *PLOS Computational Biology* 5.8 (Aug. 2009), e1000478. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000478. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000478 (visited on 11/16/2017) (cit. on pp. 9, 12).

[43] Seungryong Heo et al. “Protein Loop Structure Prediction Using Conformational Space Annealing”. In: *Journal of Chemical Information and Modeling* 57.5 (May 2017), pp. 1068–1078. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.6b00742. URL: http://dx.doi.org/10.1021/acs.jcim.6b00742 (visited on 11/16/2017) (cit. on pp. 9, 11–13).

[44] V. Collura, J. Higo, and J. Garnier. “Modeling of protein loops by simulated annealing.” In: *Protein Science : A Publication of the Protein Society* 2.9 (Sept. 1993), pp. 1502–1510. ISSN: 0961-8368. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2142460/ (visited on 05/08/2018) (cit. on pp. 9, 11, 12).

[45] James T. MacDonald, Lawrence A. Kelley, and Paul S. Freemont. “Validating a Coarse-Grained Potential Energy Function through Protein Loop Modelling”. en. In: *PLOS ONE* 8.6 (June 2013), e65770. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0065770. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0065770 (visited on 04/05/2018) (cit. on pp. 9, 11–13).

[46] C. S. Rapp and R. A. Friesner. “Prediction of loop geometries using a generalized born model of solvation effects”. eng. In: *Proteins* 35.2 (May 1999), pp. 173–183. ISSN: 0887-3585 (cit. on pp. 9, 11, 13).

[47] Viktor Hornak and Carlos Simmerling. “Generation of accurate protein loop conformations through low-barrier molecular dynamics”. fr. In: *Proteins: Structure, Function, and Bioinformatics* 51.4 (June 2003), pp. 577–590. ISSN: 1097-0134. DOI: 10.1002/prot.10363. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.10363 (visited on 05/08/2018) (cit. on p. 9).

[48] Mark A. Olson, Sidhartha Chaudhury, and Michael S. Lee. “Comparison between self-guided Langevin dynamics and molecular dynamics simulations for structure refinement of protein loop conformations”. en. In: *Journal of Computational Chemistry* 32.14 (Nov. 2011), pp. 3014–3022. ISSN: 1096-987X. DOI: 10.1002/jcc.21883. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21883 (visited on 05/08/2018) (cit. on pp. 9, 11, 13).

[49] Karina C. M. Dall’Agno and Osmar Norberto de Souza. “An expert protein loop refinement protocol by molecular dynamics simulations with restraints”. In: *Expert Systems with Applications* 40.7 (June 2013), pp. 2568–2574. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2012.10.062. URL: http://www.sciencedirect.com/science/article/pii/S0957417412011864 (visited on 05/08/2018) (cit. on pp. 9, 13).

[50] Stan Galaktionov, Gregory V. Nikiforovich, and Garland R. Marshall. “Ab initio modeling of small, medium, and large loops in proteins”. en. In: *Peptide Science* 60.2 (2001), pp. 153–168. ISSN: 1097-0282. DOI: 10.1002/1097-0282(2001)60:2<153::AID-BIP1010>3.0.CO;2-6. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0282%282001%2960%3A2%3C153%3A%3AAID-BIP1010%3E3.0.CO%3B2-6 (visited on 05/08/2018) (cit. on pp. 10, 13).

[51] Zhexin Xiang, Cinque S. Soto, and Barry Honig. “Evaluating conformational free energies: The colony energy and its application to the problem of loop prediction”. en. In: *Proceedings of the National Academy of Sciences* 99.11 (May 2002), pp. 7432–7437. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.102179699. URL: http://www.pnas.org/content/99/11/7432 (visited on 11/17/2017) (cit. on pp. 10, 12, 13).

[52] Mark A. DePristo et al. “Ab initio construction of polypeptide fragments: Efficient generation of accurate, representative ensembles”. fr. In: *Proteins: Structure, Function, and Bioinformatics* 51.1 (Apr. 2003), pp. 41–55. ISSN: 1097-0134. DOI: 10.1002/prot.10285. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.10285 (visited on 05/08/2018) (cit. on pp. 10, 12).

[53] Velin Z. Spassov, Paul K. Flook, and Lisa Yan. “LOOPER: a molecular mechanics-based algorithm for protein loop prediction”. In: *Protein Engineering, Design and Selection* 21.2 (Feb. 2008), pp. 91–100. ISSN: 1741-0126. DOI: 10.1093/protein/gzm083. URL: https://academic.oup.com/peds/article/21/2/91/1594277 (visited on 11/16/2017) (cit. on pp. 10–13).

[54] Aashish N Adhikari et al. “Modeling large regions in proteins: Applications to loops, termini, and folding”. In: *Protein Science : A Publication of the Protein Society* 21.1 (Jan. 2012), pp. 107–121. ISSN: 0961-8368. DOI: 10.1002/pro.767. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3323786/ (visited on 11/16/2017) (cit. on pp. 10–13).

[55] Shide Liang, Chi Zhang, and Yaoqi Zhou. “LEAP: Highly accurate prediction of protein loop conformations by integrating coarse-grained sampling and optimized energy scores with all-atom refinement of backbone and side chains”. en. In: *Journal of Computational Chemistry* 35.4 (Feb. 2014), pp. 335–341. ISSN: 1096-987X. DOI: 10.1002/jcc.23509. URL: http://onlinelibrary.wiley.com/doi/10.1002/jcc.23509/abstract (visited on 11/17/2017) (cit. on pp. 10–13).

[56] Ke Tang, Jinfeng Zhang, and Jie Liang. “Fast Protein Loop Sampling and Structure Prediction Using Distance-Guided Sequential Chain-Growth Monte Carlo Method”. In: *PLOS Computational Biology* 10.4 (Apr. 2014), e1003539. ISSN: 1553-7358. DOI: 10.1371/journal.

pcbi.1003539. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003539 (visited on 11/16/2017) (cit. on pp. 10, 12).

[57] Amelie Stein and Tanja Kortemme. “Improvements to Robotics-Inspired Conformational Sampling in Rosetta”. In: *PLOS ONE* 8.5 (May 2013), e63090. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0063090. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0063090 (visited on 11/18/2017) (cit. on p. 10).

[58] Ionel A. Rata, Yaohang Li, and Eric Jakobsson. “Backbone Statistical Potential from Local Sequence-Structure Interactions in Protein Loops”. In: *The Journal of Physical Chemistry B* 114.5 (Feb. 2010), pp. 1859–1869. ISSN: 1520-6106. DOI: 10.1021/jp909874g. URL: https://doi.org/10.1021/jp909874g (visited on 05/08/2018) (cit. on p. 10).

[59] Carol A. Rohl et al. “Modeling structurally variable regions in homologous proteins with rosetta”. en. In: *Proteins* 55.3 (May 2004), pp. 656–677. ISSN: 1097-0134. DOI: 10.1002/prot.10629. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.10629/abstract (visited on 11/16/2017) (cit. on pp. 10–13).

[60] Kim T. Simons et al. “Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions11Edited by F. E. Cohen”. In: *Journal of Molecular Biology* 268.1 (Apr. 1997), pp. 209–225. ISSN: 0022-2836. DOI: 10.1006/jmbi.1997.0959. URL: http://www.sciencedirect.com/science/article/pii/S0022283697909591 (visited on 05/10/2018) (cit. on p. 10).

[61] Charlotte M. Deane and Tom L. Blundell. “CODA: A combined algorithm for predicting the structurally variable regions of protein models”. In: *Protein Science : A Publication of the Protein Society* 10.3 (Mar. 2001), pp. 599–612. ISSN: 0961-8368. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2374131/ (visited on 11/16/2017) (cit. on p. 10).

[62] E. Michalsky, A. Goede, and R. Preissner. “Loops In Proteins (LIP)—a comprehensive loop database for homology modelling”. In: *Protein Engineering, Design and Selection* 16.12 (Dec. 2003), pp. 979–985. ISSN: 1741-0126. DOI: 10.1093/protein/gzg119. URL: https://academic.oup.com/peds/article/16/12/979/1513263 (visited on 11/17/2017) (cit. on p. 10).

[63] Narcis Fernandez-Fuentes, Baldomero Oliva, and András Fiser. “A supersecondary structure library and search algorithm for modeling loops in protein structures”. In: *Nucleic Acids Research* 34.7 (2006), pp. 2085–2097. ISSN: 0305-1048. DOI: 10.1093/nar/gkl156. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1440879/ (visited on 05/09/2018) (cit. on p. 10).

[64] Hung-Pin Peng and An-Suei Yang. “Modeling protein loops with knowledge-based prediction of sequence-structure alignment”. In: *Bioinformatics* 23.21 (Nov. 2007), pp. 2836–2842. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btm456. URL: https://academic.oup.com/bioinformatics/article/23/21/2836/373015 (visited on 11/16/2017) (cit. on p. 10).

[65] Yoonjoo Choi and Charlotte M. Deane. “FREAD revisited: Accurate loop structure prediction using a database search algorithm”. en. In: *Proteins: Structure, Function, and Bioinformatics* 78.6 (May 2010), pp. 1431–1440. ISSN: 1097-0134. DOI: 10.1002/prot.22658. URL: http://onlinelibrary.wiley.com/doi/10.1002/prot.22658/abstract (visited on 11/16/2017) (cit. on p. 10).

[66] Daniel Holtby, Shuai Cheng Li, and Ming Li. “LoopWeaver: Loop Modeling by the Weighted Scaling of Verified Proteins”. In: *Journal of Computational Biology* 20.3 (Mar. 2013), pp. 212–223. DOI: 10.1089/cmb.2012.0078. URL: http://online.liebertpub.com/doi/abs/10.1089/cmb.2012.0078 (visited on 11/16/2017) (cit. on pp. 10, 13).

[67] Mario Abdel Messih, Rosalba Lepore, and Anna Tramontano. “LoopIng: a template-based tool for predicting the structure of protein loops”. In: *Bioinformatics* 31.23 (Dec. 2015), pp. 3767–3772. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv438. URL: https://academic.oup.com/bioinformatics/article/31/23/3767/208242 (visited on 11/17/2017) (cit. on p. 10).

[68] Claire Marks et al. “Sphinx: merging knowledge-based and ab initio approaches to improve protein loop prediction”. In: *Bioinformatics* 33.9 (May 2017), pp. 1346–1353. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw823. URL: https://academic.oup.com/bioinformatics/article/33/9/1346/2908432 (visited on 11/17/2017) (cit. on pp. 10, 12, 13).

[69] S. P. Nguyen et al. “New Deep Learning Methods for Protein Loop Modeling”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2017), pp. 1–1. ISSN: 1545-5963. DOI: 10.1109/TCBB.2017.2784434 (cit. on p. 10).

[70] Jaume Bonet et al. “Frag’r’Us: knowledge-based sampling of protein backbone conformations for de novo structure-based protein design”. en. In: *Bioinformatics* 30.13 (July 2014), pp. 1935–1936. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btu129. URL: https://academic.oup.com/bioinformatics/article/30/13/1935/2422215 (visited on 05/13/2018) (cit. on p. 10).

[71] Robert Helling et al. “The designability of protein structures”. In: *Journal of Molecular Graphics and Modelling* 19.1 (Feb. 2001), pp. 157–167. ISSN: 1093-3263. DOI: 10.1016/S1093-3263(00)00137-6. URL: http://www.sciencedirect.com/science/article/pii/S1093326300001376 (visited on 05/10/2018) (cit. on p. 10).

[72] Mark A. Olson, Michael Feig, and Charles L. Brooks. “Prediction of protein loop conformations using multiscale modeling methods with physical energy scoring functions”. en. In: *Journal of Computational Chemistry* 29.5 (Apr. 2008), pp. 820–831. ISSN: 1096-987X. DOI: 10.1002/jcc.20827. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.20827 (visited on 05/08/2018) (cit. on pp. 11, 13).

[73] Yaohang Li, Ionel Rata, and Eric Jakobsson. “Sampling Multiple Scoring Functions Can Improve Protein Loop Structure Prediction Accuracy”. In: *Journal of chemical information and modeling* 51.7 (July 2011), pp. 1656–1666. ISSN: 1549-9596. DOI: 10.1021/ci200143u. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3211142/ (visited on 05/08/2018) (cit. on pp. 11–13).

[74] Hahnbeom Park et al. “Protein Loop Modeling Using a New Hybrid Energy Function and Its Application to Modeling in Inaccurate Structural Environments”. In: *PLOS ONE* 9.11 (Nov. 2014), e113811. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0113811. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0113811 (visited on 11/20/2017) (cit. on pp. 11–13).

[75] Ron Unger. “The Genetic Algorithm Approach to Protein Structure Prediction”. en. In: *Applications of Evolutionary Computation in Chemistry*. Structure and Bonding. Springer, Berlin,

Heidelberg, pp. 153–175. ISBN: 978-3-540-40258-9 978-3-540-44882-2. DOI: 10.1007/b13936. URL: https://link.springer.com/chapter/10.1007/b13936 (visited on 05/08/2018) (cit. on p. 11).

[76] Johannes Kästner. “Umbrella sampling”. en. In: *Wiley Interdisciplinary Reviews: Computational Molecular Science* 1.6 (Nov. 2011), pp. 932–942. ISSN: 1759-0884. DOI: 10.1002/wcms.66. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcms.66 (visited on 05/11/2018) (cit. on p. 11).

[77] Jeffrey Comer et al. “The Adaptive Biasing Force Method: Everything You Always Wanted To Know but Were Afraid To Ask”. In: *The Journal of Physical Chemistry. B* 119.3 (Jan. 2015), pp. 1129–1151. ISSN: 1520-6106. DOI: 10.1021/jp506633n. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4306294/ (visited on 05/10/2018) (cit. on p. 11).

[78] Narcis Fernandez-Fuentes, Jun Zhai, and András Fiser. “ArchPRED: a template based loop structure prediction server”. In: *Nucleic Acids Research* 34.suppl_2 (July 2006), W173–W176. ISSN: 0305-1048. DOI: 10.1093/nar/gkl113. URL: https://academic.oup.com/nar/article/34/suppl_2/W173/2505515 (visited on 11/16/2017) (cit. on p. 12).

[79] Adrian A. Canutescu and Roland L. Dunbrack. “Cyclic coordinate descent: A robotics algorithm for protein loop closure”. en. In: *Protein Science* 12.5 (May 2003), pp. 963–972. ISSN: 1469-896X. DOI: 10.1110/ps.0242703. URL: https://www.onlinelibrary.wiley.com/doi/abs/10.1110/ps.0242703 (visited on 05/08/2018) (cit. on p. 12).

[80] Peter S. Shenkin et al. “Predicting antibody hypervariable loop conformation. I. Ensembles of random conformations for ringlike structures”. en. In: *Biopolymers* 26.12 (Dec. 1987), pp. 2053–2085. ISSN: 1097-0282. DOI: 10.1002/bip.360261207. URL: http://onlinelibrary.wiley.com/doi/10.1002/bip.360261207/abstract (visited on 11/17/2017) (cit. on p. 12).

[81] Peter Minary and Michael Levitt. “Conformational Optimization with Natural Degrees of Freedom: A Novel Stochastic Chain Closure Algorithm”. In: *Journal of Computational Biology* 17.8 (Aug. 2010), pp. 993–1010. DOI: 10.1089/cmb.2010.0016. URL: https://www.liebertpub.com/doi/full/10.1089/cmb.2010.0016 (visited on 05/08/2018) (cit. on p. 12).

[82] Evangelos A. Coutsias et al. “A kinematic view of loop closure”. en. In: *Journal of Computational Chemistry* 25.4 (Mar. 2004), pp. 510–528. ISSN: 1096-987X. DOI: 10.1002/jcc.10416. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.10416 (visited on 05/08/2018) (cit. on p. 12).

[83] William J. Wedemeyer and Harold A. Scheraga. “Exact analytical loop closure in proteins using polynomial equations”. en. In: *Journal of Computational Chemistry* 20.8 (June 1999), pp. 819–844. ISSN: 1096-987X. DOI: 10.1002/(SICI)1096-987X(199906)20:8<819::AID-JCC8>3.0.CO;2-Y. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291096-987X%28199906%2920%3A8%3C819%3A%3AAID-JCC8%3E3.0.CO%3B2-Y (visited on 05/08/2018) (cit. on p. 12).

[84] Federico Fogolari and Silvio C. E. Tosatto. “Application of MM/PBSA colony free energy to loop decoy discrimination: Toward correlation between energy and root mean square deviation”. en. In: *Protein Science* 14.4 (Apr. 2005), pp. 889–901. ISSN: 1469-896X. DOI: 10.1110/ps.041004105. URL: https://onlinelibrary.wiley.com/doi/abs/10.1110/ps.041004105 (visited on 05/08/2018) (cit. on p. 13).

[85] Yuedong Yang and Yaoqi Zhou. “Specific interactions for ab initio folding of protein terminal regions with secondary structures”. en. In: *Proteins: Structure, Function, and Bioinformatics* 72.2 (Aug. 2008), pp. 793–803. ISSN: 1097-0134. DOI: 10.1002/prot.21968. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21968 (visited on 05/08/2018) (cit. on p. 13).

[86] Hongyi Zhou and Jeffrey Skolnick. “GOAP: A Generalized Orientation-Dependent, All-Atom Statistical Potential for Protein Structure Prediction”. In: *Biophysical Journal* 101.8 (Oct. 2011), pp. 2043–2052. ISSN: 0006-3495. DOI: 10.1016/j.bpj.2011.09.012. URL: http://www.sciencedirect.com/science/article/pii/S0006349511010708 (visited on 05/08/2018) (cit. on p. 13).

[87] Amarda Shehu, Cecilia Clementi, and Lydia E. Kavraki. “Modeling protein conformational ensembles: From missing loops to equilibrium fluctuations”. en. In: *Proteins: Structure, Function, and Bioinformatics* 65.1 (Oct. 2006), pp. 164–179. ISSN: 1097-0134. DOI: 10.1002/

prot.21060. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21060 (visited on 05/08/2018) (cit. on p. 15).

[88] Jerome Nilmeier et al. “Assessing protein loop flexibility by hierarchical Monte Carlo sampling”. In: *Journal of chemical theory and computation* 7.5 (May 2011), pp. 1564–1574. ISSN: 1549-9618. DOI: 10.1021/ct1006696. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3129859/ (visited on 05/10/2018) (cit. on p. 15).

[89] Claire Marks, Jiye Shi, and Charlotte M. Deane. “Predicting loop conformational ensembles”. eng. In: *Bioinformatics (Oxford, England)* 34.6 (Mar. 2018), pp. 949–956. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btx718 (cit. on p. 15).

[90] Amarda Shehu, Cecilia Clementi, and Lydia E. Kavraki. “Sampling Conformation Space to Model Equilibrium Fluctuations in Proteins”. en. In: *Algorithmica* 48.4 (Aug. 2007), pp. 303–327. ISSN: 0178-4617, 1432-0541. DOI: 10.1007/s00453-007-0178-0. URL: https://link.springer.com/article/10.1007/s00453-007-0178-0 (visited on 05/08/2018) (cit. on p. 15).

[91] Noah C. Benson and Valerie Daggett. “Dynameomics: Large-scale assessment of native protein flexibility”. en. In: *Protein Science* 17.12 (Dec. 2008), pp. 2038–2050. ISSN: 1469-896X. DOI: 10.1110/ps.037473.108. URL: https://onlinelibrary.wiley.com/doi/abs/10.1110/ps.037473.108 (visited on 05/12/2018) (cit. on p. 15).

[92] Donald J. Jacobs et al. “Protein flexibility predictions using graph theory”. en. In: *Proteins: Structure, Function, and Bioinformatics* 44.2 (Aug. 2001), pp. 150–165. ISSN: 1097-0134. DOI: 10.1002/prot.1081. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.1081 (visited on 05/12/2018) (cit. on p. 15).

[93] B. P. Pandey et al. “Protein flexibility prediction by an all-atom mean-field statistical theory”. en. In: *Protein Science* 14.7 (July 2005), pp. 1772–1777. ISSN: 1469-896X. DOI: 10.1110/ps.041311005. URL: https://onlinelibrary.wiley.com/doi/abs/10.1110/ps.041311005 (visited on 05/12/2018) (cit. on p. 15).

[94] Sara E. Dobbins, Victor I. Lesk, and Michael J. E. Sternberg. “Insights into protein flexibility: The relationship between normal modes and conformational change upon protein–protein

docking”. en. In: *Proceedings of the National Academy of Sciences* 105.30 (July 2008), pp. 10390–10395. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0802496105. URL: http://www.pnas.org/content/105/30/10390 (visited on 05/12/2018) (cit. on p. 15).

[95] Ranja Sarkar. “Native flexibility of structurally homologous proteins: insights from anisotropic network model”. In: *BMC Biophysics* 10 (Jan. 2017), p. 1. ISSN: 2046-1682. DOI: 10.1186/s13628-017-0034-9. URL: https://doi.org/10.1186/s13628-017-0034-9 (visited on 05/12/2018) (cit. on p. 15).

[96] David Bramer and Guo-Wei Wei. “Multiscale weighted colored graphs for protein flexibility and rigidity analysis”. In: *The Journal of Chemical Physics* 148.5 (Feb. 2018), p. 054103. ISSN: 0021-9606. DOI: 10.1063/1.5016562. URL: https://aip.scitation.org/doi/10.1063/1.5016562 (visited on 04/12/2018) (cit. on p. 15).

Chapter 2

Remodeling the protein backbone to position a catalytic residue

Natural proteins often use highly structured loops to position key functional residues. Mimicking this approach in rationally designed proteins is an important step towards the routine design of functional proteins. But even though it is now often possible to predict loop structures with sub-angstrom accuracy, designing structured loops to position functional residues with similar accuracy remains an unsolved problem. We are approaching this problem by developing a protocol called Pull Into Place (PIP) that iterates between flexible-backbone design and state-of-the-art loop structure prediction. The first step of our protocol searches for backbones that support the desired sidechain geometry by using loop modeling simulations with gentle restraints to hold the sidechains in place. The second step finds sequences that stabilize the backbone models found in the first step. The third step uses unrestrained loop modeling simulations to eliminate designs for which the desired structure is not the lowest in energy. Good models from the third step are then fed back into the second step for further optimization. We are testing our protocol by attempting to rescue the Asp-to-Glu mutation of the catalytic residue in bacterial ketosteroid-isomerase (KSI), which moves the catalytic carboxylate by 1.8Å RMSD and causes a 240-fold decrease in k_{cat}. Although we have successfully rescued the positioning of the catalytic carboxylate, we have not yet been able to rescue the enzyme's catalytic activity. Our hope is that further development of this pipeline will yield significant progress towards the design of functional proteins.

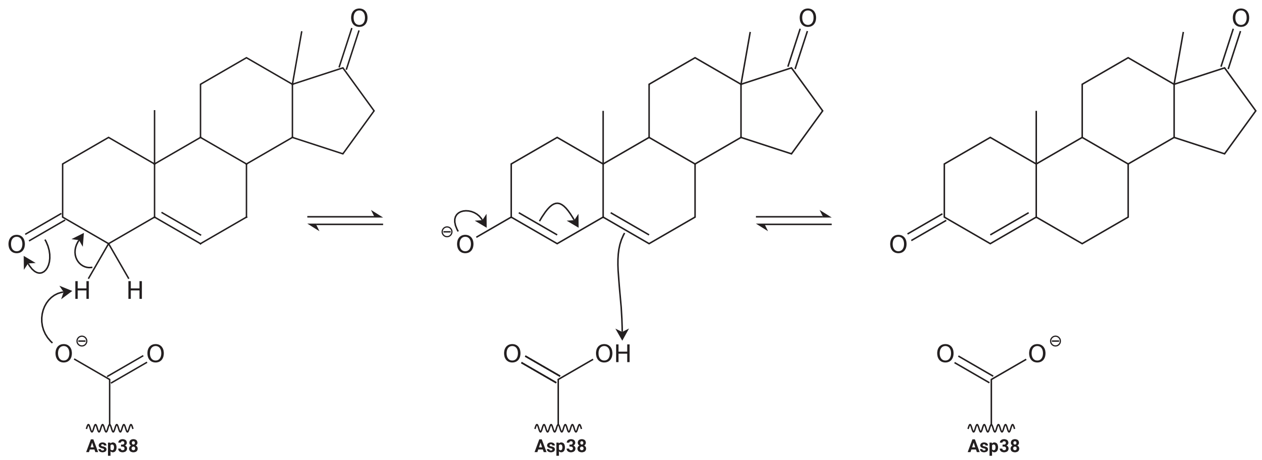

Figure 2.1: Mechanism of the isomerization catalyzed by KSI.

2.1 Model system

Ketosteroid isomerase (KSI) is an enzyme that catalyzes the rearrangement of a double bond in steroid molecules. The mechanism is outlined in Figure 2.1. Briefly, the general base (D38) abstracts a proton to form an enolate intermediate, which then collapses to form the enone product. The reaction naturally progresses in the forward direction due to the increased conjugation of the enone compared to the isolated ketone and alkene groups. Note that the first step involves a carboxylate ($pK_a = 4.5$) [1] deprotonating an allylic carbon ($pK_a = 12.7$) [2], which would normally be prohibitively unfavorable. It occurs in the KSI active site due to the influence of the oxyanion hole (Figure 2.2).

The precise positioning of the D38 carboxylate group has a strong effect on catalysis. The D38E mutant, which moves the carboxylate by 1.8Å RMSD*, decreases k_{cat} by 240-fold [3]. Double-mutant cycle experiments have established which residues are most important for accurately positioning the carboxylate [3, 4]. The first these residues are F54 and F116: the two phenylalanines flanking D38 [3]. Although they seem to bury the carboxylate in a hydrophobic pocket, they are actually engaging it in two anion-aromatic interactions. Anion-aromatic interactions are driven by the fact that electron density is localized in the center of aromatic rings, creating a partial positive charge around the edge of the ring that attracts negatively charged residues†. The third residue

*From superimposing 4L7K on 8CHO.

†This is the opposite of the cation-π interaction, where for the same reason positively charged residues are attracted to the face of aromatic rings.

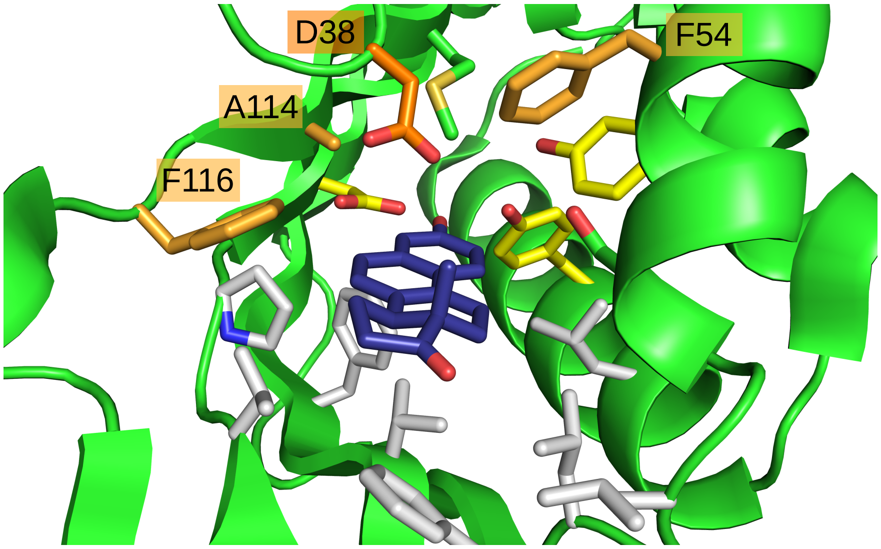

Figure 2.2: Important residues in the KSI active site. Purple: substrate analog equilenin. Orange: the general base (D38). Pale orange: residues that have been shown to contribute to the positioning of D38. Yellow: the oxyanion hole. White: the hydrophobic binding pocket.

important for positioning D38 is A114, simply because it packs against the D38 sidechain [4]. The rest of the residues (P39, V40, G41) make up the active site loop, and are collectively the most important contributors [4].

Our goal is to rescue the catalysis of the D38E mutant by redesigning the active site loop to position the E38 carboxylate in the same place as the wildtype D38 carboxylate (Figure 2.3). Our primary interest is to develop a method for very accurately remodeling structured loops, and there are a number of features that make KSI a good model system for such a project. First is that by starting from an enzyme that's highly functional in every way except the one we deliberately perturbed, we control for many of the challenges associated with enzyme design in general and focus on the challenges associated with remodeling the backbone. Second is that KSI has been well studied. In addition to the double mutant cycles described above, there is a wealth of information available on how catalysis is affected by the oxyanion hole [5] and the electrostatic field created by the enzyme [6]. Third is that KSI is a small and soluble protein. Fourth is that the active site loop is tolerant to mutation. It has been shown that the entire loop can be mutated (to glycine) with no apparent effect on K_M, suggesting that the active site loop is not involved in either binding the substrate or stabilizing the protein fold [4]. Fifth is that k_{cat} is very sensitive to small changes in carboxylate positioning, as evidenced by the 240-fold response to the 1.8Å perturbation made by

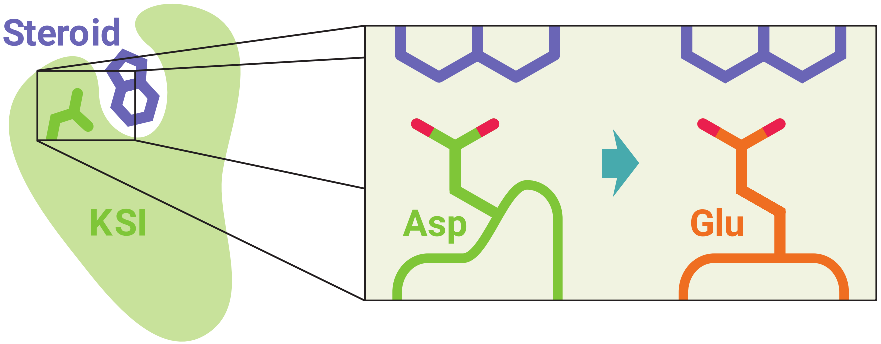

Figure 2.3: Our goal is the redesign the active site of KSI to use a glutamic acid for catalysis, rather than an aspartic acid.

the D38E mutation. This gives us a way to make sub-angstrom inferences about the position of the active site loop by just measuring k_{cat}, which we can do in higher throughput than we can solve crystal structures. The 240-fold effect on k_{cat} also gives us a large dynamic range in which to observe improvements made by our designs. Sixth is that established assays exist to measure KSI's catalytic activity, and seventh is that KSI has proven very amenable to structure determination via both x-ray crystallography and nuclear magnetic resonance (NMR), so we stand a good chance of being able to visualize our designs.

Despite all those positives, there are some drawbacks to using KSI as a model system. First is that the enzyme assay requires purified protein, which limits the number of designs we can test. Second is that enzymes are very sensitive, and there are a number of ways we could incidentally harm catalysis that are unrelated to the positioning of the carboxylate group. For example, we could alter electrostatic interactions (which are long-range, if not effectively shielded by solvent), or the pK_a of various residues, or solvent access to the active site, or the dynamics of the designed loop, or something else. Third is that the anion-aromatic interaction that helps stabilize the carboxylate is not explicitly modeled by the Rosetta score function. As a result, the energy of the desired conformation may be be overestimated in our simulations.

2.2 Results

To address the challenge of remodeling a protein backbone to position one or more functionally important sidechains with sub-angstrom accuracy, we developed a new computational protocol

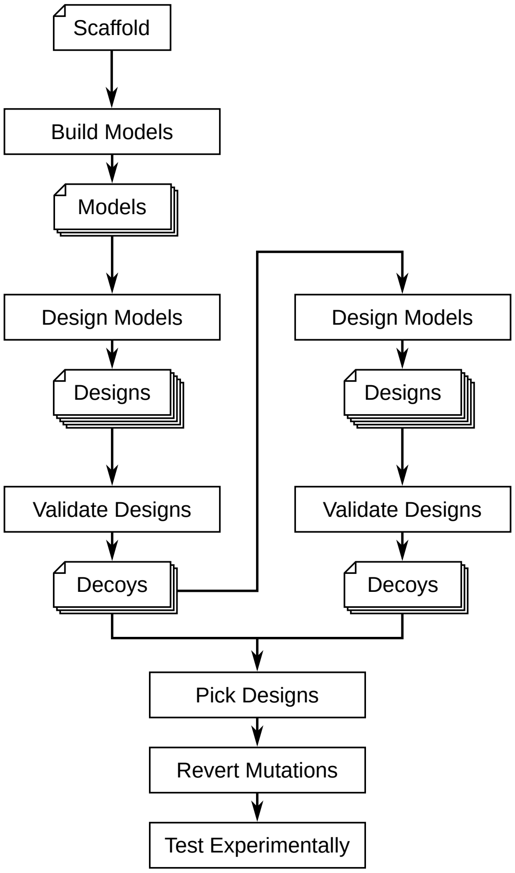

Figure 2.4: Schematic of the PIP protocol.

called Pull Into Place (PIP). PIP seeks to build on the the kinematic closure (KIC) algorithm, which is often successful at predicting sub-angstrom loop conformations [7, 8], in a three-step process (Figure 2.4). The first step ("build models") is to use KIC to generate backbone conformations capable of satisfying the design goal, which in this case is to position the E38 sidechain where it can catalyze the reaction. The design goal is represented using harmonic restraints, so the simulation can simultaneously optimize for conformations that satisfy the restraints and are physically realistic. The second step ("design models") is to design optimized sequences for each backbone, without sampling new backbone conformations. The third step ("validate designs") is to use KIC again, this time without any restraints, to predict the preferred conformations of the designed sequences. Sequences that are predicted to prefer their designed conformation are candidates for experimental testing. The third step typically generates a number of models that satisfy the design goals well despite scoring poorly. These models are fed back into the second step, to see if their scores could be improved by another round of sequence optimization. In this way, the second and third steps can be iterated.

As a proof of concept, we applied the PIP protocol to the problem of remodeling the active site loop in KSI to rescue the D38E mutant. We independently created designs for two different loop lengths: the first the same length as the wildtype loop (13 residues), and the second with a one-residue deletion (12 residues). We began the protocol by creating 4,128 models that positioned the E38 such that all three carboxylate atoms were within 0.6Å of their intended positions (Table 2.1). From these models we designed 176,417 unique sequences to stabilize the remodeled backbone and E38 sidechain conformations (Table 2.2). We chose 200 of these sequences to validate computationally. For 41 of the designs, the predicted structure places all three of the E38 carboxylate atoms within 1.2Å of their intended positions (Table 2.3). These designs were said to "pass validation". We also found 300 models (mostly with one deletion in the loop) that positioned the carboxylate atoms within 0.6Å of their intended positions. We used these to design 12,825 more unique (Table 2.2), of which 100 were validated and 70 passed (Table 2.3). Note that the fraction of designs that passed validation increased from 21% in the first round to 70% in the second round, suggesting that backbone models from unrestrained loop modeling simulations are good scaffolds for design. As described in the Methods, we visually inspected the 41+70=111 designs that passed validation and picked 14 to test experimentally (Table 2.6). We then attempted simplify

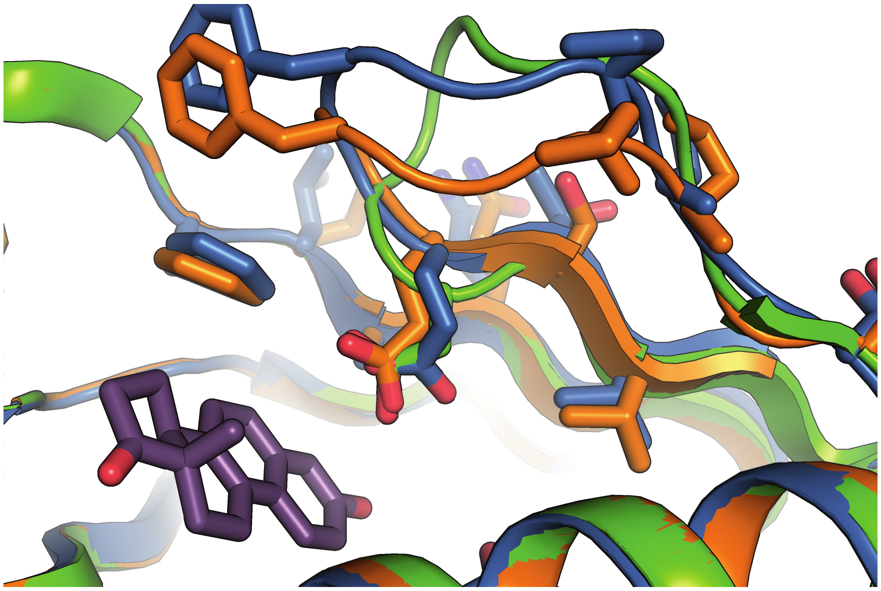

Figure 2.5: Comparison of wildtype KSI (green), the design L model (orange), and the design L crystal structure (blue). The crystal structure was solved by Lin Liu.

each designs by computationally validating different combinations of wildtype reversion mutations (Table 2.7).

Of the 14 designs we tested experimentally, 11 could be expressed. None were soluble, but 10 could be purified from inclusion bodies. Only 4 of these were stable, and 3 had detectable enzymatic activity in the assay described in [4] (Table 2.4). We calculated Michaelis-Menten parameters for design L (Table 2.5, Figure 2.6a). From this we can conclude that design L actually made catalysis worse than the D38E mutant alone, but did not affect substrate binding. Despite the poor activity, we solved the crystal structure for this design (Figure 2.5). Surprisingly, the geometry matched the design model fairly well. The carboxylate group in the crystal structure was offset 1.28Å C/O/O RMSD from the model. The Cα was even closer: just 0.61Å from the model. The RMSD between the crystal structure and design model for the 12-residue remodeled loop was 1.41Å, mostly due to small hinge motions in S39 and F44.

However, the crystal structure does not explain the loss in catalysis, since E38 is positioned better than in design L than in the D38E mutant. Instead, the loss in catalysis is better explained by the fact that the design is a monomer in solution (Figure 2.6b). The design is a dimer in the crystal structure, which suggests that our design might adopt a different conformation in solution that it does in the structure. Wildtype KSI is not active as a monomer, so being monomeric could

# Dels	Loop	# Models	<0.6Å
0	34–46	10340	1943
1	34–45	9998	2185

Table 2.1: Models capable of correctly positioning E38. # Dels: The number of deletions in the active site loop. Loop: The residues comprising the active site loop. # Models: The total number of backbone models that were generated. <0.6Å: The number of backbone models that positioned all three E38 carboxylate atoms within 0.6Å of their intended positions.

Round	# Dels	# Inputs	# Designs	# Unique
1	0	1943	96689	84853
1	1	2185	108929	91564
2	0	24	7150	4190
2	1	276	13754	8635

Table 2.2: Designs that stabilize the correct positioning of E38. Round: The first or second iteration of the design step. # Dels: The number of deletions in the active site loop. # Inputs: The number of backbone models that were used as the scaffold for fixed-backbone design. # Designs: The total number of designs that were for generated. # Unique: The number of unique designs that were generated.

Round	# Dels	# Designs	<1.2Å	# Picked
1	0	100	21	4
1	1	100	21	3
2	0	50	38	3
2	1	50	32	4

Table 2.3: Computationally validated designs. Round: The first or second iteration of the validation step. # Dels: The number of deletions in the active site loop. # Designs: The number of designs that were chosen for computational validation. <1.2Å: The number of designs for which the lowest scoring decoy (of 500) positioned all three atoms of the E38 carboxylate within 1.2Å of their intended positions. # Picked: The number of designs that were picked for experimental testing.

Name	Expressed	Soluble	Purified	Stable	Active
A	✓		✓	✓	
B	✓		✓		
C					
D	✓				
E	✓		✓		
F	✓		✓		
G	✓		✓		
H					
I	✓		✓		
J	✓		✓		
K	✓		✓	✓	✓
L	✓		✓	✓	✓
M	✓		✓	✓	✓
N					

Table 2.4: Experimental validation of the KSI designs. Name: The name of the design. Expressed: Whether or not the design was expressed, as indicated by its presence in either the soluble of insoluble fraction (determined by visual inspection of a Coomassie PAGE gel). Soluble: Whether or not the design was soluble, as indicated by its presence in the soluble fraction (determined as above). Purified: Whether or not the design could be purified. Designs that could be expressed but were not soluble were purified from inclusion bodies (see Methods). Stable: Whether or not the design remained soluble after being stored at 4°C for 1 day. Active: Whether or not the design catalyzed the isomerization of 5(10)-estrene-3,17-dione above the baseline rate (determined by the visual inspection of an absorbance vs. time plot for a single enzyme concentration). All data in this table was collected by Lin Liu.

Name	k_{cat} (s^{-1})	K_M (µM)
wildtype	36	50
D38E	0.15	35
design L	0.0027	38

Table 2.5: Michaelis-Menten parameters for design L. The data for design L was collected by Lin Liu. The data for wildtype KSI and the D38E mutant are reproduced from [4] for comparison.

explain the poor activity of the design. Changes in the quaternary structure of the scaffold were also not accounted for by our computational models. Initial efforts to restore the dimeric quaternary structure by making reversion mutations have so far been unsuccessful.

2.3 Discussion

The PIP protocol presented in this chapter is neither a beginning nor an end. Older versions incorporated ideas that we later thought better of, and newer versions have attempted to address the shortcomings of their predecessors. The most recent version of the protocol is available from: http://pull-into-place.readthedocs.io/en/latest/. In the interest of informing future efforts to develop PIP (or similar protocols), here we will discuss the reasoning behind both the ideas we moved away from, and the ideas we are moving toward.

2.3.1 Build models

Our first consideration was how to create backbone models capable of supporting desired sidechain geometries. Before we settled on using flexible backbone simulations with harmonic restraints, we experimented with using inverse rotamers instead. In hindsight, restraints are the better option for a number of reasons. The idea of inverse rotamers is to hold the end of a sidechain fixed in some desired position, then to rotate the sidechain torsions to create a library of rotamers that all have their backbone atoms in different positions. The challenge is then to connect the backbone for the rest of protein with the various inverse rotamers. We generated inverse rotamers that positioned the E38 carboxylate exactly as in wildtype KSI, then used KIC close the two resulting gaps in the backbone: one from the N-terminal edge of the loop to E38, and another from E38 to the C-terminal edge of the loop. In principle, closing a gap requires setting at least 6 torsions, or 3 residues (since each residue has φ and ψ torsions). In practice, there often aren't any solutions if only 3 residues can move, so at least 4 are needed. This brings us to the first problem with inverse rotamers: you have to move 4 residues on either side of each residue you have inverse rotamers for. This would make it impossible to design two residues that are separated by less than that[‡]. It also means

[‡] For example, the two residues responsible for recognizing the protospacer adjacent motif (PAM) in Cas9 are separated by only one residue, and thus could not be designed with this approach.

that a loop designed to position one residue can be no shorter than 9 residues long (4 on each side, and 1 for the residue itself), which is already a nontrivial length, although for KSI we chose to design more residues than that anyway. The second problem with inverse rotamers is that the resulting models, which have the sidechain positioned perfectly and the backbone fully connected, are not realistic. By forcing the sidechain into the perfect position, we also force the backbone into a relatively strained conformation. When the backbone is allowed to move freely, it often adopts a different conformation and doesn't position the sidechain correctly. Using restraints lets the simulation balance the strain between the sidechain and the backbone and favors the models that minimize that strain the most.

We are now attempting to apply more powerful backbone sampling methods to the model building step, specifically fragment KIC (fKIC) or loop hash KIC (lhKIC) (Listing 2.8). Both algorithms incorporate structural information from the protein data bank (PDB) in their sampling. fKIC compares the sequence of the region being sampled to structural motifs derived from the PDB, then samples backbone torsions from the motifs that are most similar in sequence. lhKIC instead searches for structural motifs from the PDB that nearly connect the ends of the region being sampled, then samples both backbone torsions and residue identities from those motifs. A significant advantage sampling from the PDB is the ability to design secondary structure. In the context of the KSI project, this means that both algorithms sometimes produce models that expand the β-sheet adjacent to the active site loop. For comparison, the designs described in this chapter were modeled with NGK, which samples backbone torsions from the two-body Ramachandran distribution. While this incorporates correlations between adjacent residues, it did not produce models with expanded secondary structure.

One factor that makes lhKIC (at present) more conceptually pleasing is that it also samples sequence. Although the primary focus of the model building step is to sample backbone conformations, we allow the sequence to change simultaneously because it doesn't make sense to sample new loop conformations exclusively in the context of the wildtype sequence. This is a problem for fKIC, which uses a static database of structural motifs selected based on their similarity to the wildtype sequence. As the simulation progresses and the sequence accumulates more and more mutations, that database gets more and more out-of-date. lhKIC avoids this problem by taking both sequence and structure from the motifs it finds. We note that fKIC could address this problem by picking fragments from a dynamically updating database, but this feature does not yet exist.

We are also now attempting to remodel larger stretches of the backbone in the model building step (Listing 2.15,2.16). For example, we often strive to begin and end remodeling in secondary structural elements, where we expect the structure to be robust and reliable. To make this possible, we keep all the initial coordinates of the scaffold, instead of discarding the coordinates for the loop (Listing 2.8). This reduces the space of structures that needs to be searched, and immediately focuses on those loops that are most similar to the wildtype (and therefore the most plausible) while still fulfilling the design goals.

2.3.2 Design models

Depending on whether a KSI is considered successful based on its crystal structure or its enzymatic activity, we had either 1 or 0 successful designs in 14 tries. One probable reason for this low success rate is that we were far too liberal in both which positions we allowed to design and which amino acids we allowed those positions to design to (Listing 2.3,2.7). In 125- and 124-residue scaffolds (corresponding to the two loop lengths), we allowed 41 and 38 residues to mutate. Making this many mutations severely stresses a fundamental assumption in protein design, which is that the conformation of the scaffold won't significantly change. In our current efforts to remodel the active site loop in KSI, we have been more conservative about which positions are allowed to design. Not counting E38, we chose 15 positions to mutate by visually inspecting the structure . We also are using LayerDesign ConsensusLoopDesign to limit the allowed amino acid identities based on the burial, secondary structure, and turn geometry of each position (Listing 2.11) [9].

When subjected to computational validation, a surprising number of KSI designs were predicted to adopt the right backbone geometry but the wrong E38 rotamer. One way to address this issue and to increase the number of designs that pass validation is to consider both positive and negative conformations in the design step. Specifically, after iterating between design and validation a few times, we often have competitive off-target models for each design. Subsequent rounds of design could use one of the multi-state design paradigms in Rosetta [10, 11, 12] to selectively destabilize those off-target states.

We're also trying to improve the design models step by using flexible backbone design rather than fixed backbone design. The goal is to allow the backbone to move enough to accommodate all plausible sidechain rotamers, but not enough to significantly change conformation. The protocol we're currently using for this purpose is FastDesign (Listing 2.9), which works by iterating between sidechain rotamer optimization and gradient minimization while ramping the temperature between iterations [13]. The backbone moves slightly because we allow the torsions within the loop to minimize. We have also considered using CoupledMoves [14] as an alternative to FastDesign.

2.3.3 Validate designs

Because we can produce so many more designs (≈200,000) than we can computationally test (≈200), picking which designs to validate is an important task. For the designs described in this chapter, we approached this task in two ways. The first was to simply pick the designs with the lowest scores. This turned out to be inefficient because the lowest scoring designs tended to have very similar sequences (i.e. differing only in inconsequential positions). Repeatedly validating variants of one or two motifs was a waste of our computational resources. Our second approach was to pick designs with a probability proportional to their Boltzmann-weighted score. In other words, we sought to prefer designs with low scores, but not to the exclusion of designs with slightly higher scores. This gave us a better diversity of designs to test.

The drawback to both of the design-picking methods described above is that they only consider score. We are now scoring each design by a variety of metrics, including score, buried unsatisfied H-bonds, E38 rotamer probability, E38 interaction score, exposed hydrophobic surface area, predicted pK_a, and fragment quality (Listing 2.12). We then leverage the information in these metrics by picking the designs that are on the Pareto front. Simply put, a design is on the Pareto front if no other design scores better than it in every metric. This will include the designs that score the best in each individual metric, along with those that score generally well in multiple metrics. Since designs that score well for different reasons tend to have different sequences, this is an excellent way to pick a diverse set of designs to validate.

To get more accurate validation results, we have switched to using fKIC instead of NGK for these simulations (Listing 2.10). In addition to being a better algorithm for sampling secondary structure, fKIC also substantially outperforms NGK on a difficult 16-residue loop modeling benchmark (Roland Pache & Xingjie Pan, personal communication). We are also experimenting with validating 10x more designs by performing 10x fewer fKIC simulations for each design. For loop modeling applications, it's necessary to run 500 simulations per loop in order to get accurate predictions for the most difficult loops in our benchmarks. However, other loops can be successfully predicted in just a handful of simulations. For design applications, it may be smarter to focus on the latter kind of loop. We're currently testing whether 50 simulations is enough to identify designs worth testing. Interestingly, with this few simulations, the fragment generation step required by fKIC becomes to be limiting in terms of both time and space. If we built a way to efficiently pick fragments on-the-fly, it might be worth testing as few as 10 fKIC simulations per design.

2.4 Conclusion

We have developed a computational protocol to automate the design of structured loops. Broadly speaking, the protocol has three steps: building models that satisfy the design goal, designing optimized sequences for those models, and computationally validating the designs via loop modeling. In the process of developing this protocol, we gained valuable practical experience on the topic of structured loop design. Although we have not yet executed a successful design, we have promising structural results suggesting that this protocol is capable of designing structured loops.

2.5 Methods

2.5.1 Computational design

Rosetta

The designs described in this chapter were generated using Rosetta version 10b6f2f8e20d70757e6b510def2ddcbeef172538 with the talaris2013 score function.

Input Files

There were two structures of KSI that we could have based our designs on: 8CHO [15] and 1QJG [16]. 8CHO has an empty binding pocket, while 1QJG is binding a substrate analog (equilenin) and has the D38N mutation. Both structures have the same resolution (2.3Å), R_{work} (0.205), and R_{free} (0.271). We chose to start from 1QJG, since we thought it would be valuable to have the ligand in the structure. Even though the active site loop doesn't interact with the ligand directly, we wanted to avoid creating designs that inadvertently occlude the ligand-binding pocket. We created parameters for the equilenin ligand using the `molfile_to_params.py` script distributed with Rosetta (Listing 2.1,2.2). KSI is an obligate dimer, so we included both monomers in our initial structure. To design two different loop lengths, we created two versions of the initial structure: one with a deletion and one without. Finally, we replaced N38 with E38 by hand and relaxed the resulting models in the talaris2013 score function using FastRelax, with only sidechain degrees-of-freedom allowed to move.

The desired position of the E38 sidechain was expressed in a restraint file (Listing 2.5). Each atom in the E38 carboxylate group was restrained to the position of the corresponding atom in the N38 amide group in the structure we began from (1QJG). We visually confirmed that N38 in 1QJG has the same rotameric conformation as D38 in 8CHO.

The residues being remodeled were expressed in loop files (Listing 2.6,2.7). There are two loop files because we designed two different loop lengths. We chose which residues to remodel using our intuition. Our goals were to provide adequate room for remodeling on either side of E38, while minimizing loop length to improve the accuracy of loop modeling. The loops we chose were 13 and 12 residue long. For comparison, NGK has been shown to predict 12-residue loops with sub-angstrom accuracy [8].

The residues that were allowed to design and repack were specified in a resfile (Listing 2.3,2.4). There are two resfiles because we designed two different loop lengths, and changing the loop length changes the indices of all the other residues in the protein. Any residue that had a sidechain atom within 4Å or 6Å of any loop atom in any model generated in the "Build models" step was allowed to design or repack, respectively. F54, A114, and F116 were not allowed to design because they are known to be important for positioning the catalytic residue. Each designed residue was allowed become any of the 20 canonical amino acids except cysteine (due to the potential for disulfide bonds) and histidine (due to the potential for pH-dependent behavior).

Note that the inputs described above are out-of-date. As we've continued to work on remodeling the KSI active site loop, we've changed which algorithms we use, which atom we restrain, which loops we remodel, which residues we design and repack, etc. The most recent input files are available from the following repository: https://github.com/Kortemme-Lab/ksi_inputs.git. Furthermore, the specific input files used in this chapter (including the PDB files) and listed in the appendix can be found in the `3967a341318008b2c614ba43164d5c82bc0f50b1` commit of this repository, in the subdirectories labeled "v1" (where relevant).

Build models

We created models satisfying the design goal by running 10,000 next generation KIC [8] simulations (Listing 2.17) with restraints as described above (Listing 2.5). Backbone remodeling was limited to the loop defined in the appropriate file (Listing 2.6,2.7) and design was allowed according the appropriate resfile (Listing 2.3,2.4). The initial coordinates of the loop being remodeled were discarded and rebuilt from scratch. Only models that put all three restrained atoms within 0.6Å of their intended positions were carried on to the next step.

Design models

We used fixed-backbone design to stabilize models that correctly positioned E38. We ran 50 fixbb simulations per model, or more if there were relatively few models (Listing 2.18). Design was allowed according the appropriate resfile (Listing 2.3,2.4). We picked 50–100 designs to validate with probability proportional to their Boltzmann-weighted talaris2013 scores (in REU).

Validate models

We computationally validated our designs by running 500 NGK simulations for each one (Listing 2.19). Backbone movement was limited to the loop defined in the appropriate file (Listing 2.6,2.7). The initial coordinates for that loop were discarded and rebuilt from scratch. Any design for which the lowest scoring decoy put all three carboxylate atoms within 1.2Å of their intended positions was carried on to the manual screening step. Furthermore, any decoy at all (regardless of score) that put all three carboxylate atoms within 0.6Å of their intended positions was used as input for a second round of design simulations.

Manual screening

We picked designs to experimentally validate by comparing quality metrics and visually inspecting models. The quality metrics are described in Table 2.6. We paid particular attention to the score gap and the number of buried unsatisfied H-bonds. We also made an effort to pick designs from different clusters. We visually inspected the lowest scoring model for each design to eliminate those with unreasonable backbone or sidechain conformations.

Wildtype reversions

For each design selected for experimental validation, we reran the simulations described in the “Validate models” subsection for each individual wildtype reversion mutation. We then combined any reversions that had no apparent effect and reran the validation simulations again. In cases where the combination of all the individually acceptable reversions had a deleterious effect, we manually picked more conservative combinations of reversions to validate. If no acceptable combination of reversions could be found, no reversions were made (Table 2.7).

2.5.2 Experimental validation

The 14 design chosen for experimental validation were ordered from GenScript pre-cloned into the pET-21a expression vector. Unless otherwise noted, the following protocols are contributed by Lin Liu:

Expression

1. Start a 2 L LB broth culture from an overnight culture (1:100 dilution) with 50 µg/mL carbenicillin

2. Grow to an O.D. of 0.6 (~4 hours) at 37°C. Remove 1 mL aliquot of cells before induction for SDS-PAGE.
3. Induce with 0.5 mM IPTG (1 mL of 1 M IPTG in 2 L LB medium)
4. Grow another 3-4 hours, then remove 1 mL aliquot of cells for SDS-PAGE (post-induction sample).
5. Harvest cells by centrifugation (3500 rpm/20'/4°C)
6. Resuspend in 10-15 mL of lysis buffer: (40 mM Tris-HCl, 1 mM EDTA, 25% sucrose w/v, pH 8.5) Remove 100 µL aliquot for SDS-PAGE.
7. Freeze (unless using immediately)

Harvesting inclusion bodies

1. Thaw cells (if frozen above)
2. Lyse using emulsiflex (4–5 passes with air pressure knob set to 60–80)
3. Centrifuge at 20,000 rpm/20'/4°C in JA-20 rotor with Oakridge tubes.
4. Discard supernatant
5. Solubilize pellet in ~20-25 mL of 20 mM Tris-HCl, 1% sodium deoxycholate, 200 mM NaCl, 2 mM EGTA, pH 8.5 using a dounce. (Can try vortexing the pellet with buffer first before resorting to dounce)
6. Centrifuge at 8,000g/10'/4°C in JA-20 (Oakridge tubes) or conical tube centrifuge (with falcon tubes)
7. Remove 100 µL aliquot of supernatant for SDS-PAGE, then discard rest of supernatant
8. Solubilize pellet in 25 mL of 10 mM Tris-HCl, 0.25% sodium deoxycholate, pH 8.5
9. Centrifuge at 8,000g/5'/4°C.
10. Discard supernatant.

11. Repeat steps 8-10 at least 2x, or until supernatant clear.

12. Solubilize in 25 mL of 20 mM Na-HEPES, 500 mM NaCl, 1 mM EDTA, pH 8.5 to remove detergent.

13. Centrifuge at 8,000g/5’/4°C.

14. Discard supernatant.

15. Repeat steps 12-14 at least 2x, or until supernatant clear.

16. Freeze purified inclusion bodies (unless using immediately)

Solubilizing and refolding inclusion bodies

1. Add 10 mL 8 M urea, 20 mM Na-HEPES, 500 mM NaCl, 1 mM EDTA, pH 8.5, 10 mM DTT (add fresh DTT). Incubate on shaker for 30’ to dissolve, or use dounce.

2. Centrifuge at 20,000 rpm/20’/4°C in JA-20 (Oakridge), or 15,000 rpm/15’/4°C in conical tube centrifuge (falcon tubes)

3. Add supernatant to 200 mL 40 mM KPi, 1 mM EDTA, 2 mM DTT while stirring at 4°C. Continue stirring for 1 hour (0.4 M urea final)

4. Filter with 0.4 µm filter (can use 250 mL filter bottle, or syringes. May have to add filter paper on top of bottle attachment to prevent clogs).

5. Save 100 µL aliquot of refolded material for SDS-PAGE.

Purification

1. Pre-equilibrate 5-10 mL deoxycholate affinity column with 40 mM KPi, 1 mM EDTA, 2 mM DTT, pH 7.2

2. Apply filtered refolded enzyme to deoxycholate column by gravity at 4°C. Can use large funnel attached to falcon tube funnel, covered with saran wrap. Will take overnight or longer. Save 100 µL aliquot of flow-through for SDS-PAGE.

3. 3. Wash with 100 mL 400 mM KPi, 1 mM EDTA, 2 mM DTT, pH 7.2 with vacuum manifold.

4. Wash with 50 mL 40 mM KPi, 1 mM EDTA, 2 mM DTT, pH 7.2 with vacuum manifold.

5. Elute with 25 mL 40 mM KPi, 1 mM EDTA, 50% EtOH, 2 mM DTT, pH 7.2. Collect 1 mL fractions, and check using mini-Bradford for protein.

6. Pool fractions that contain protein (If < 6 mL, adjust to 6 mL volume with 50% EtOH buffer).

7. Load on superose 12 size exclusion column:

    (a) Isocratic 1 mL/min 40 mM KPi, 1 mM EDTA, 2 mM DTT

    (b) 6 mL injection

    (c) 128 mL total flow

    (d) Collect 2 mL fractions after 40 min (so after 40 mL flow-through).

8. Pool fractions containing protein (by absorbance).

9. Concentrate in Amicon Ultra 10 kDa cutoff spin filter units at 3500 rpm/20'/4°C.

10. Store concentrated enzyme at 4°C.

Concentration determination

1. Determine protein concentration by recording (A280 - A320) for a series of dilutions in 40 mM KPi, 6 M GuHCl, pH 7, and using $\text{□} = 16860\,\mathrm{M}^{-1}\mathrm{cm}^{-1}$ for WT pKSI (use `http://www.basic.northwestern.edu/biotools/proteincalc.html` for other mutants).

2. Typical protocol:

    (a) Goal is to get all absorbance values between 0.05–1 (preferably 0.1–1).

    (b) Prepare 1:10 dilution of stock protein in 40 mM KPi.

    (c) Prepare 0, 10, 20, 30, 40% dilutions of above stock in 40 mM KPi

    (d) Take 25 µL of each dilution, and add 75 µL of 40 mM KPi, 8 M GuHCl, pH 7.

    (e) Record absorbance spectra from 240 – 400 nm, blanking with 25 µL 40 mM KPi + 75 µL 40 mM KPi, 8 M GuHCl.

(f) For an original stock concentration of 2.5 mM WT pKSI, this method will give A280 ~ 0.1 for the lowest protein sample (1:400 net dilution). If protein of interest has smaller ε or expected concentration, adjust dilutions accordingly to get A280 between 0.05–1.

Check purity

1. 5% Tris-glycine SDS-PAGE (37.5:1 acrylamide:bis-acrylamide)
2. Before loading, heat all samples (except the MW marker) at 95°C for 5 min.

Typical lanes:

1. Molecular weight (MW) marker
2. Pre-induction
   - (a) Centrifuge 1 mL cells aliquot at 14,000g/5'/4°C.
   - (b) Resuspend cells in 100 µL of 5x SB, and heat at 95°C for 5 min
   - (c) Load 10 µL of resuspended cells
3. Post-induction:
   - (a) Prepare as above.
   - (b) Load 10 µL.
4. Lysate supernatant
   - (a) Centrifuge 100 µL cell lysate at 14,000g/5'/4°C.
   - (b) Transfer 100 µL of supernatant into fresh tube.
   - (c) Load 5 µL of this supernatant + 5 µL 5x SB.
5. Lysate pellet
   - (a) Resuspend pellet from part (d) above in 100 µL of 20 mM NaPi, 7 M urea, pH 7.2 (or any high urea buffer)
   - (b) Load 5 µL of this suspension + 5 µL 5x SB.

6. 1% DOC wash:
   (a) Load 5 µL of DOC wash + 5 µL 5x SB.
7. Refolded enzyme, before affinity step:
   (a) Load 15 µL + 5 µL 5x SB (note large loading volume due to huge dilution here).
8. h. Affinity flow-through:
   (a) Load 15 µL + 5 µL 5x SB (again note large loading volume).
9. Final protein
   (a) If the concentration of the final protein is known, loading 10 µL of 50–100 µM pure protein solution (0.5–1 nmol) should be more than sufficient to see plenty of material by Coomassie without drastic overloading for a 10-lane gel. Adjust volume accordingly for higher concentrations.
   (b) If the concentration is unknown, 1–5 mM pure protein solution after final concentration step is typical. Loading 1 µL of 1 mM solution should more than suffice, and loading 1 µL of 2 mM or greater solutions can result in noticeable overloading of lanes. For a publication quality gel, it is probably best to quantify amount of protein first

Activity assay

KSI activity was determined as in [4].

Crystallography

Crystallization buffer: 1.0M $(NH_4)_2SO_4$. 2 µL buffer, 2 µL protein (3.5 $^{mg}/_{mL}$). CRYST1 52.570 52.570 177.910 90.00 90.00 120.00 P 65 2 2. Resolution: 1.97Å. $R_{\text{work}} = 0.2658$. $R_{\text{free}} = 0.3159$.

2.6 Appendix

2.6.1 Input Files

Listing 2.1: Centroid-mode parameters for the equilenin ligand.

```
NAME EQU
IO_STRING EQU Z
TYPE LIGAND
AA UNK
ATOM  C10 CAbb  X   -0.09
ATOM  C4  CAbb  X   -0.09
ATOM  C3  CAbb  X   -0.09
ATOM  C2  CAbb  X   -0.09
ATOM  C1  CAbb  X   -0.09
ATOM  O1  OCbb  X   -0.63
ATOM  H1  HNbb  X   0.46
ATOM  C6  CAbb  X   -0.09
ATOM  C5  CAbb  X   -0.09
ATOM  C7  CAbb  X   -0.09
ATOM  C8  CAbb  X   -0.09
ATOM  C9  CAbb  X   -0.09
ATOM  C11 CAbb  X   -0.06
ATOM  C12 CAbb  X   -0.06
ATOM  C13 CAbb  X   -0.15
ATOM  C14 CAbb  X   -0.15
ATOM  C17 CAbb  X   0.65
ATOM  C16 CAbb  X   -0.15
ATOM  C15 CAbb  X   -0.15
ATOM  O2  OCbb  X   -0.73
ATOM  C18 CAbb  X   -0.24
BOND_TYPE  C1   O1  1
BOND_TYPE  C1   C2  4
BOND_TYPE  C1   C6  4
BOND_TYPE  O1   H1  1
BOND_TYPE  C2   C3  4
BOND_TYPE  C3   C4  4
BOND_TYPE  C3   C7  4
BOND_TYPE  C4   C5  4
BOND_TYPE  C4   C10 4
BOND_TYPE  C5   C6  4
BOND_TYPE  C7   C8  4
BOND_TYPE  C8   C9  4
BOND_TYPE  C9   C10 4
BOND_TYPE  C9   C11 1
BOND_TYPE  C10  C14 1
BOND_TYPE  C11  C12 1
BOND_TYPE  C11  C15 1
BOND_TYPE  C12  C13 1
BOND_TYPE  C12  C17 1
BOND_TYPE  C12  C18 1
BOND_TYPE  C13  C14 1
BOND_TYPE  C15  C16 1
BOND_TYPE  C16  C17 1
BOND_TYPE  C17  O2  2
CHI 1  C2   C1   O1   H1
PROTON_CHI 1 SAMPLES 2 0 180 EXTRA 1 20
NBR_ATOM  C10
NBR_RADIUS 6.457284
ICOOR_INTERNAL    C10    0.000000    0.000000    0.000000   C10   C4    C3
ICOOR_INTERNAL    C4     0.000000  180.000000    1.418226   C10   C4    C3
ICOOR_INTERNAL    C3     0.000000   60.891172    1.489702   C4    C10   C3
ICOOR_INTERNAL    C2  -179.653613   59.343553    1.418490   C3    C4    C10
ICOOR_INTERNAL    C1     0.206150   62.634983    1.387028   C2    C3    C4
ICOOR_INTERNAL    O1   179.723817   63.181116    1.286966   C1    C2    C3
ICOOR_INTERNAL    H1     0.131327   60.000772    0.936981   O1    C1    C2
ICOOR_INTERNAL    C6  -179.865025   58.144838    1.432274   C1    C2    O1
ICOOR_INTERNAL    C5    -0.054123   58.255220    1.379395   C6    C1    C2
ICOOR_INTERNAL    C7   179.848594   58.760349    1.444112   C3    C4    C2
ICOOR_INTERNAL    C8    -0.451279   61.728926    1.438722   C7    C3    C4
ICOOR_INTERNAL    C9     0.133493   59.739449    1.438572   C8    C7    C3
ICOOR_INTERNAL    C11 -178.865124   61.637379    1.475424   C9    C8    C7
ICOOR_INTERNAL    C12 -152.758365   67.129337    1.548286   C11   C9    C8
ICOOR_INTERNAL    C13  -54.167776   68.113044    1.540645   C12   C11   C9
ICOOR_INTERNAL    C14   52.841508   67.456988    1.555093   C13   C12   C11
ICOOR_INTERNAL    C17 -124.747589   83.656618    1.511318   C12   C11   C13
ICOOR_INTERNAL    C16  -31.441534   69.687007    1.576369   C17   C12   C11
ICOOR_INTERNAL    C15   -0.413699   79.661791    1.576716   C16   C17   C12
ICOOR_INTERNAL    O2   179.877290   56.474680    1.205589   C17   C12   C16
ICOOR_INTERNAL    C18 -110.972689   67.126267    1.544635   C12   C11   C17
```

Listing 2.2: Fullatom-mode parameters for the equilenin ligand.

```
NAME EQU
IO_STRING EQU Z
TYPE LIGAND
AA UNK
ATOM  C10 aroC  X   -0.09
ATOM  C4  aroC  X   -0.09
ATOM  C3  aroC  X   -0.09
ATOM  C2  aroC  X   -0.09
ATOM  C1  aroC  X   -0.09
ATOM  O1  OH    X   -0.63
ATOM  H1  Hpol  X   0.46
ATOM  C6  aroC  X   -0.09
ATOM  C5  aroC  X   -0.09
ATOM  H3  Haro  X   0.14
ATOM  H4  Haro  X   0.14
ATOM  H2  Haro  X   0.14
ATOM  C7  aroC  X   -0.09
ATOM  C8  aroC  X   -0.09
ATOM  C9  aroC  X   -0.09
ATOM  C11 CH1   X   -0.06
ATOM  C12 CH1   X   -0.06
ATOM  C13 CH2   X   -0.15
ATOM  C14 CH2   X   -0.15
ATOM  H10 Hapo  X   0.12
ATOM  H11 Hapo  X   0.12
ATOM  H8  Hapo  X   0.12
ATOM  H9  Hapo  X   0.12
ATOM  C17 COO   X   0.65
ATOM  C16 CH2   X   -0.15
ATOM  C15 CH2   X   -0.15
ATOM  H12 Hapo  X   0.12
ATOM  H13 Hapo  X   0.12
ATOM  H14 Hapo  X   0.12
ATOM  H15 Hapo  X   0.12
ATOM  O2  OOC   X   -0.73
ATOM  C18 CH3   X   -0.24
ATOM  H16 Hapo  X   0.12
ATOM  H17 Hapo  X   0.12
ATOM  H18 Hapo  X   0.12
ATOM  H7  Hapo  X   0.12
ATOM  H6  Haro  X   0.14
ATOM  H5  Haro  X   0.14
BOND_TYPE  C1   O1  1
BOND_TYPE  C1   C2  4
BOND_TYPE  C1   C6  4
BOND_TYPE  O1   H1  1
BOND_TYPE  C2   C3  4
BOND_TYPE  C2   H2  1
BOND_TYPE  C3   C4  4
BOND_TYPE  C3   C7  4
BOND_TYPE  C4   C5  4
BOND_TYPE  C4   C10 4
BOND_TYPE  C5   C6  4
BOND_TYPE  C5   H3  1
BOND_TYPE  C6   H4  1
BOND_TYPE  C7   C8  4
BOND_TYPE  C7   H5  1
BOND_TYPE  C8   C9  4
BOND_TYPE  C8   H6  1
BOND_TYPE  C9   C10 4
BOND_TYPE  C9   C11 1
BOND_TYPE  C10  C14 1
BOND_TYPE  C11  C12 1
BOND_TYPE  C11  C15 1
BOND_TYPE  C11  H7  1
BOND_TYPE  C12  C13 1
BOND_TYPE  C12  C17 1
BOND_TYPE  C12  C18 1
BOND_TYPE  C13  C14 1
BOND_TYPE  C13  H8  1
BOND_TYPE  C13  H9  1
BOND_TYPE  C14  H10 1
BOND_TYPE  C14  H11 1
BOND_TYPE  C15  C16 1
BOND_TYPE  C15  H12 1
BOND_TYPE  C15  H13 1
BOND_TYPE  C16  C17 1
BOND_TYPE  C16  H14 1
BOND_TYPE  C16  H15 1
BOND_TYPE  C17  O2  2
BOND_TYPE  C18  H16 1
BOND_TYPE  C18  H17 1
BOND_TYPE  C18  H18 1
CHI 1  C2   C1   O1   H1
PROTON_CHI 1 SAMPLES 2 0 180 EXTRA 1 20
NBR_ATOM  C10
NBR_RADIUS 6.457284
ICOOR_INTERNAL    C10    0.000000    0.000000    0.000000   C10   C4    C3
ICOOR_INTERNAL    C4      0.000000  180.000000    1.418226   C10   C4    C3
ICOOR_INTERNAL    C3      0.000000   60.891172    1.489702   C4    C10   C3
ICOOR_INTERNAL    C2   -179.653613   59.343553    1.418490   C3    C4    C10
ICOOR_INTERNAL    C1      0.206150   62.634983    1.387028   C2    C3    C4
ICOOR_INTERNAL    O1    179.723817   63.181116    1.286966   C1    C2    C3
ICOOR_INTERNAL    H1      0.131327   60.000772    0.936981   O1    C1    C2
ICOOR_INTERNAL    C6   -179.865025   58.144838    1.432274   C1    C2    O1
ICOOR_INTERNAL    C5     -0.054123   58.255220    1.379395   C6    C1    C2
ICOOR_INTERNAL    H3   -179.816263   59.664315    1.031977   C5    C6    C1
ICOOR_INTERNAL    H4   -179.999559   60.871901    1.032042   C6    C1    C5
ICOOR_INTERNAL    H2   -179.997784   58.684437    1.031961   C2    C3    C1
ICOOR_INTERNAL    C7    179.848594   58.760349    1.444112   C3    C4    C2
ICOOR_INTERNAL    C8     -0.451279   61.728926    1.438722   C7    C3    C4
ICOOR_INTERNAL    C9      0.133493   59.739449    1.438572   C8    C7    C3
ICOOR_INTERNAL    C11 -178.865124   61.637379    1.475424   C9    C8    C7
ICOOR_INTERNAL    C12 -152.758365   67.129337    1.548286   C11   C9    C8
ICOOR_INTERNAL    C13  -54.167776   68.113044    1.540645   C12   C11   C9
ICOOR_INTERNAL    C14   52.841508   67.456988    1.555093   C13   C12   C11
ICOOR_INTERNAL    H10   94.989030   72.130472    1.070042   C14   C13   C12
ICOOR_INTERNAL    H11  122.345167   74.240083    1.069964   C14   C13   H10
ICOOR_INTERNAL    H8  -120.456750   71.300890    1.070002   C13   C12   C14
ICOOR_INTERNAL    H9  -121.039781   72.329013    1.070015   C13   C12   H8
ICOOR_INTERNAL    C17 -124.747589   83.656618    1.511318   C12   C11   C13
ICOOR_INTERNAL    C16  -31.441534   69.687007    1.576369   C17   C12   C11
ICOOR_INTERNAL    C15   -0.413699   79.661791    1.576716   C16   C17   C12
ICOOR_INTERNAL    H12  150.943087   68.073569    1.069962   C15   C16   C17
ICOOR_INTERNAL    H13  117.878279   64.756126    1.069995   C15   C16   H12
ICOOR_INTERNAL    H14  118.503132   68.296876    1.070046   C16   C17   C15
ICOOR_INTERNAL    H15  117.995988   65.280179    1.069947   C16   C17   H14
ICOOR_INTERNAL    O2    179.877290   56.474680    1.205589   C17   C12   C16
ICOOR_INTERNAL    C18 -110.972689   67.126267    1.544635   C12   C11   C17
ICOOR_INTERNAL    H16  -75.495035   70.531520    1.070031   C18   C12   C11
ICOOR_INTERNAL    H17 -120.001823   70.529126    1.069978   C18   C12   H16
ICOOR_INTERNAL    H18 -119.999702   70.532272    1.069991   C18   C12   H17
ICOOR_INTERNAL    H7  -123.199482   82.078338    1.069972   C11   C9    C12
ICOOR_INTERNAL    H6  -179.999034   60.128775    1.031975   C8    C7    C9
ICOOR_INTERNAL    H5   179.998938   59.135188    1.031969   C7    C3    C8
```

Listing 2.3: Resfile for the input model with no deletions.

```
NATRO
START

# Design residues in the loop itself.  Don't move the catalytic residue,
# because we want to find designs which stabilize that rotamer.

34 - 37     A NOTAA HC
38          A NATRO
39 - 46     A NOTAA HC

# Design any residue that has a sidechain atom within 4A of any loop atom in
# any input model.  Phe54, Ala114, and Phe116 are excluded because they are
# known to be important for positioning the catalytic residue.

30          A NOTAA HC
31          A NOTAA HC
32          A NOTAA HC
33          A NOTAA HC
47          A NOTAA HC
48          A NOTAA HC
49          A NOTAA HC
50          A NOTAA HC
51          A NOTAA HC
52          A NOTAA HC
53          A NOTAA HC
55          A NOTAA HC
57          A NOTAA HC
```

```
58          A NOTAA HC
60          A NOTAA HC
109         A NOTAA HC
110         A NOTAA HC
111         A NOTAA HC
112         A NOTAA HC
113         A NOTAA HC
115         A NOTAA HC
117         A NOTAA HC
118         A NOTAA HC
121         A NOTAA HC
199         B NOTAA HC
200         B NOTAA HC
201         B NOTAA HC
202         B NOTAA HC

# Repack any residue that has a sidechain atom within 6A of any loop atom in
# any input model.

10          A NATAA
11          A NATAA
13          A NATAA
14          A NATAA
15          A NATAA
16          A NATAA
17          A NATAA
18          A NATAA
23          A NATAA
26          A NATAA
27          A NATAA
29          A NATAA
54          A NATAA
56          A NATAA
```

59	A NATAA
61	A NATAA
63	A NATAA
77	A NATAA
78	A NATAA
79	A NATAA
80	A NATAA
81	A NATAA
82	A NATAA
84	A NATAA
86	A NATAA
93	A NATAA
95	A NATAA
96	A NATAA
97	A NATAA
98	A NATAA
99	A NATAA
100	A NATAA
101	A NATAA
102	A NATAA
103	A NATAA
104	A NATAA
108	A NATAA
114	A NATAA
116	A NATAA
119	A NATAA
120	A NATAA
122	A NATAA
123	A NATAA
127	B NATAA
129	B NATAA
132	B NATAA
197	B NATAA

```
198         B NATAA
203         B NATAA
204         B NATAA
205         B NATAA
224         B NATAA
225         B NATAA
226         B NATAA
227         B NATAA
228         B NATAA
234         B NATAA
235         B NATAA
236         B NATAA
237         B NATAA
238         B NATAA
239         B NATAA
```

Listing 2.4: Resfile for the input model with one deletion.

```
NATRO
START

# Design residues in the loop itself.  Don't move the catalytic residue,
# because we want to find designs which stabilize that rotamer.

34 - 37     A NOTAA HC
38          A NATRO
39 - 45     A NOTAA HC

# Design any residue that has a sidechain atom within 4A of any loop atom in
# any input model.  Phe53, Ala113, and Phe115 are excluded because they are
# known to be important for positioning the catalytic residue.

29          A NOTAA HC
30          A NOTAA HC
```

```
31          A NOTAA HC
32          A NOTAA HC
33          A NOTAA HC
46          A NOTAA HC
48          A NOTAA HC
49          A NOTAA HC
50          A NOTAA HC
52          A NOTAA HC
54          A NOTAA HC
56          A NOTAA HC
57          A NOTAA HC
108         A NOTAA HC
109         A NOTAA HC
110         A NOTAA HC
111         A NOTAA HC
112         A NOTAA HC
114         A NOTAA HC
116         A NOTAA HC
117         A NOTAA HC
120         A NOTAA HC
198         B NOTAA HC
199         B NOTAA HC
200         B NOTAA HC
201         B NOTAA HC

# Repack any residue that has a sidechain atom within 6A of any loop atom in
# any input model.

10          A NATAA
11          A NATAA
13          A NATAA
14          A NATAA
15          A NATAA
```

16	A NATAA
17	A NATAA
18	A NATAA
23	A NATAA
26	A NATAA
27	A NATAA
28	A NATAA
47	A NATAA
51	A NATAA
53	A NATAA
55	A NATAA
58	A NATAA
59	A NATAA
60	A NATAA
62	A NATAA
76	A NATAA
77	A NATAA
78	A NATAA
79	A NATAA
80	A NATAA
81	A NATAA
83	A NATAA
85	A NATAA
92	A NATAA
94	A NATAA
95	A NATAA
96	A NATAA
97	A NATAA
98	A NATAA
99	A NATAA
100	A NATAA
101	A NATAA
102	A NATAA

```
103         A NATAA
107         A NATAA
113         A NATAA
115         A NATAA
118         A NATAA
119         A NATAA
121         A NATAA
122         A NATAA
126         B NATAA
128         B NATAA
131         B NATAA
196         B NATAA
197         B NATAA
202         B NATAA
203         B NATAA
204         B NATAA
223         B NATAA
224         B NATAA
225         B NATAA
226         B NATAA
227         B NATAA
233         B NATAA
234         B NATAA
235         B NATAA
236         B NATAA
237         B NATAA
238         B NATAA
```

Listing 2.5: Harmonic restraints reflecting the design goal

```
CoordinateConstraint OE1 38 CA 1 17.895 73.085 10.634 HARMONIC 0.0 1.0
CoordinateConstraint OE2 38 CA 1 19.471 74.505 10.507 HARMONIC 0.0 1.0
CoordinateConstraint CG  38 CA 1 20.090 72.256 10.794 HARMONIC 0.0 0.707
```

Listing 2.6: Loop positions for the input model with no deletions.

```
LOOP 34 46 46 0 1
```

Listing 2.7: Loop positions for the input model with one deletion.

```
LOOP 34 45 45 0 1
```

Listing 2.8: The most recent version of the ”build models” step.

```
<ROSETTASCRIPTS>

  {% include "shared_defs.xml" %}

  <TASKOPERATIONS>
    <RestrictToRepacking name="repackonly"/>
  </TASKOPERATIONS>

  <MOVERS>
    <LoopModeler name="modeler"
      config="loophash_kic"
      scorefxn_fa="scorefxn_cst"
      task_operations="resfile,repackonly,ex,aro,curr"
      loops_file="{{ w.loops_path }}"
      loophash_perturb_sequence="yes"
      loophash_seqposes_no_mutate="38"
      fast="{{ 'yes' if test_run else 'no' }}"
    />
  </MOVERS>

  <PROTOCOLS>
    <!-- Constraints read from command line -->
    <Add mover_name="modeler"/>
    <Add mover_name="writer"/>
  </PROTOCOLS>

  <OUTPUT scorefxn="scorefxn"/>
```

```
</ROSETTASCRIPTS>
```

Listing 2.9: The most recent version of the ”design models” step.

```
<ROSETTASCRIPTS>

  {% include "shared_defs.xml" %}

  <RESIDUE_SELECTORS>
    <Index name="turn" resnums="199-200"/>
  </RESIDUE_SELECTORS>

  <TASKOPERATIONS>
    <LayerDesign name="layer"
        ignore_pikaa_natro="yes"/>
    <ConsensusLoopDesign name="abego"
        residue_selector="turn"
        include_adjacent_residues="no"/>
  </TASKOPERATIONS>

  <MOVERS>
    <AtomTree name="foldtree" fold_tree_file="{{ w.find_path('foldtree') }}"/>
    <AtomTree name="unfoldtree" simple_ft="yes"/>
    <AddChainBreak name="break_loop" resnum="39" change_foldtree="no"/>
    <AddChainBreak name="break_turn" resnum="199" change_foldtree="no"/>
    <FastDesign name="fastdesign"
        task_operations="resfile,layer,abego,ex,aro,curr"
        scorefxn="scorefxn_cst" >
      <MoveMap bb="no" chi="yes" jump="no">
         <Span begin="26"  end="50"  chi="yes" bb="yes"/>
         <Span begin="197" end="202" chi="yes" bb="yes"/>
      </MoveMap>
    </FastDesign>
  </MOVERS>

  <PROTOCOLS>
    <Add mover_name="nativebonus"/>
    <Add mover_name="cst"/> <!-- Added via mover b/c command-line ignored. -->
    <Add mover_name="foldtree"/>
    <Add mover_name="break_loop"/>
    <Add mover_name="break_turn"/>
    <Add mover_name="fastdesign"/>
    <Add mover_name="unfoldtree"/> <!-- Otherwise Foldability segfaults. -->
    <Add mover_name="writer"/>
  </PROTOCOLS>

  <OUTPUT scorefxn="scorefxn"/>

</ROSETTASCRIPTS>
```

Listing 2.10: The most recent version of the ”validate designs” step.

```
<ROSETTASCRIPTS>

  {% include "shared_defs.xml" %}

  <MOVERS>
    <LoopModeler name="modeler"
      config="kic_with_frags"
      scorefxn_fa="scorefxn"
      loops_file="{{ w.loops_path }}"
      fast="{{ 'yes' if test_run else 'no' }}">
        <Build skip="yes"/>
    </LoopModeler>
  </MOVERS>

  <PROTOCOLS>
```

```
    <Add mover_name="modeler"/>
    <Add mover_name="writer"/>
  </PROTOCOLS>

  <OUTPUT scorefxn="scorefxn"/>

</ROSETTASCRIPTS>
```

Listing 2.11: Shared RosettaScript elements for the above steps.

```
  {% include "filters.xml" %}

  <SCOREFXNS>
    <ScoreFunction name="scorefxn" weights="{{ w.scorefxn_path }}"/>
    <ScoreFunction name="scorefxn_cst" weights="{{ w.scorefxn_path }}">
      <Reweight scoretype="coordinate_constraint" weight="1.0"/>
      <Reweight scoretype="atom_pair_constraint" weight="1.0"/>
      <Reweight scoretype="angle_constraint" weight="1.0"/>
      <Reweight scoretype="dihedral_constraint" weight="1.0"/>
      <Reweight scoretype="res_type_constraint" weight="1.0"/>
      <Reweight scoretype="chainbreak" weight="100.0"/>
    </ScoreFunction>
  </SCOREFXNS>

  <RESIDUE_SELECTORS>
    <Chain name="chA" chains="A"/>
    <Index name="E38" resnums="38"/>
  </RESIDUE_SELECTORS>

  <TASKOPERATIONS>
    <ReadResfile name="resfile"/>
    <ExtraRotamersGeneric name="ex" ex1="yes" ex2="yes" extrachi_cutoff="0"/>
    <LimitAromaChi2 name="aro" include_trp="yes"/>
    <IncludeCurrent name="curr"/>
  </TASKOPERATIONS>

  <MOVERS>
    <FavorNativeResidue name="nativebonus" />
    <ConstraintSetMover name="cst" cst_fa_file="{{ w.restraints_path }}"/>
    <WriteFiltersToPose name="writer" prefix="EXTRA_METRIC "/>
  </MOVERS>
```

Listing 2.12: Score metrics calculated in the above steps.

```
<FILTERS>
  <PackStat
    name="PackStat Score [+]"
    threshold="0"
    chain="0"
    repeats="1"
  />
  <ResidueIE
    name="E38 Interaction Energy [-|REU]"
    scorefxn="scorefxn_cst"
    score_type="total_score"
    energy_cutoff="-10"
    restype3="GLU"
    interface="0"
    whole_pose="0"
    selector="E38"
    jump_number="1"
    interface_distance_cutoff="8.0"
    max_penalty="1000.0"
    penalty_factor="1.0"
  />
  <PreProline
    name="Pre-Proline Potential [-]"
    use_statistical_potential="true"
```

```
/>
<TotalSasa
  name="Total SASA [-|Å²]"
  threshold="0"
  upper_threshold="1000000000000000"
  hydrophobic="0"
  polar="0"
/>
<ExposedHydrophobics
  name="Exposed Hydrophobic Residue SASA [-|Å²]"
  sasa_cutoff="20"
  threshold="-1"
/>
<HbondsToResidue
  name="H-bonds to E38 [+|#]"
  scorefxn="scorefxn_cst"
  partners="0"
  energy_cutoff="-0.5"
  backbone="true"
  bb_bb="true"
  sidechain="true"
  residue="38"
  from_other_chains="true"
  from_same_chain="true"
/>
<HbondsToResidue
  name="H-bonds to E38 (Backbone) [+|#]"
  scorefxn="scorefxn_cst"
  partners="0"
  energy_cutoff="-0.5"
  backbone="true"
  bb_bb="true"
  sidechain="false"
  residue="38"
  from_other_chains="true"
  from_same_chain="true"
/>
<HbondsToResidue
  name="H-bonds to E38 (Sidechain) [+|#]"
  scorefxn="scorefxn_cst"
  partners="0"
  energy_cutoff="-0.5"
  backbone="false"
  bb_bb="false"
  sidechain="true"
  residue="38"
  from_other_chains="true"
  from_same_chain="true"
/>
<BuriedUnsatHbonds
  name="Buried Unsatisfied H-Bonds [-|#]"
  scorefxn="scorefxn"
  print_out_info_to_pdb="true"
  task_operations="resfile"
/>
<OversaturatedHbondAcceptorFilter
  name="Oversaturated H-bonds [-|#]"
  scorefxn="scorefxn_cst"
  max_allowed_oversaturated="0"
  hbond_energy_cutoff="-0.5"
  consider_mainchain_only="false"
/>
<RepackWithoutLigand
  name="Repack Without Ligand Δ[-|REU]"
  scorefxn="scorefxn_cst"
  target_res="all_repacked"
  rms_threshold="100"
/>
{% if w.focus_name != 'validate_designs' %}
<Foldability
  name="Foldability (35-41) [+]"
  tries="60"
  start_res="35"
  end_res="41"
/>
<Foldability
  name="Foldability (37-44) [+]"
  tries="60"
  start_res="37"
  end_res="44"
/>
{% endif %}
<FragmentScoreFilter name="Max 9-Residue Fragment αC RMSD [-|Å]"
  scoretype="FragmentCrmsd"
  sort_by="FragmentCrmsd"
  threshold="9999"
  direction="-"
  start_res="{{ w.largest_loop.start }}"
  end_res="{{ w.largest_loop.end }}"
  compute="maximum"
  outputs_folder="{{ w.seqprof_dir }}"
  outputs_name="%%job_id%%"
  csblast="/netapp/home/krivacic/software/csblast-2.2.3_linux64"
  blast_pgp="/netapp/home/klabqb3backrub/tools/blast-2.2.26/bin/blastpgp"
  placeholder_seqs="/netapp/home/xingjiepan/Databases/BLAST/placeholder/placeholder_seqs"
  psipred="/netapp/home/xingjiepan/Softwares/parametric_scaffold_design/dependencies/depe
  sparks-x="/netapp/home/klabqb3backrub/tools/sparks-x"
  sparks-x_query="/netapp/home/klabqb3backrub/tools/sparks-x/bin/buildinp_query.sh"
  frags_scoring_config="{{ w.fragment_weights_path }}"
```

```
    n_frags="200"
    n_candidates="1000"
    fragment_size="9"
    vall_path="{{ w.rosetta_vall_path(test_run) }}"
  />
</FILTERS>
```

Listing 2.13: Most recent resfile, for the input model with no deletions.

```
NATRO
START

# Design positions
# ================
# This resfile simply encompasses the whole active site loop, the beta
# strand leading up to it, and the turn on the opposite monomer.  Compared to
# our previous resfile, this one allows two more positions to design on each
# side of the active site loop, which we hope will lead to design models that
# better satisfy the constraints:
#
# Positions 35+37: Previously we excluded these positions because they're
# pointing into solvent and clearly not affecting the active site loop.
# Despite that, we now think that the more important thing is to create a more
# diverse set of backbones in the initial model building step.  Allowing
# loophash to design the whole loop should help accomplish that.  We'll use
# LayerDesign in the next step to make sure we still end up with greasy
# residues on the inside and polar one on the outside.
#
# Positions 45+46: Design the active site loop up to G47.  We don't think it
# would be wise to include G47 in the design, since it seems to be responsible
# for breaking the α-helix at position 48.  Previously we stopped at position
# 44 in hopes of keeping its salt-bridge with E53, but Rosetta usually got rid
# of that interaction anyways.
#
```

```
# We'll be using LayerDesign and ConsensusLoopDesign to restrict which residues
# can go where, so the only thing we're specifying here is `NOTAA CH`.  We
# never want cysteine because it can mess up the global fold by forming
# unexpected disulfides, and we never want histidine because it's behavior can
# be pH dependent.

34      A NOTAA CH
35      A NOTAA CH
36      A NOTAA CH
37      A NOTAA CH
38      A PIKAA E
39      A NOTAA CH
40      A NOTAA CH
41      A NOTAA CH
42      A NOTAA CH
43      A NOTAA CH
44      A NOTAA CH
45      A NOTAA CH
46      A NOTAA CH
199     B NOTAA CH
200     B NOTAA CH
201     B NOTAA CH
202     B NOTAA CH

# Repack positions
# ================
# The following repack positions were chosen by the clash-based repack
# shell creator (excluding the ligand).

14      A NATAA
30      A NATAA
50      A NATAA
51      A NATAA
```

```
54      A NATAA
55      A NATAA
95      A NATAA
109     A NATAA
111     A NATAA
112     A NATAA
113     A NATAA
114     A NATAA
115     A NATAA
116     A NATAA
121     A NATAA
127     B NATAA
204     B NATAA
225     B NATAA
227     B NATAA

# The following repack positions were added after visual inspection of
# clash-based repack shell.

10      A NATAA
13      A NATAA
17      A NATAA
25      A NATAA
52      A NATAA
53      A NATAA
56      A NATAA
57      A NATAA
58      A NATAA
108     A NATAA
110     A NATAA
117     A NATAA
118     A NATAA
126     B NATAA
```

```
128     B NATAA
228     B NATAA
```

Listing 2.14: Most recent resfile, for the input model with one deletion.

```
NATRO
START

# Design positions
# ================
# This resfile simply encompasses the whole active site loop, the beta
# strand leading up to it, and the turn on the opposite monomer.  Compared to
# our previous resfile, this one allows two more positions to design on each
# side of the active site loop, which we hope will lead to design models that
# better satisfy the constraints:
#
# Positions 35+37: Previously we excluded these positions because they're
# pointing into solvent and clearly not affecting the active site loop.
# Despite that, we now think that the more important thing is to create a more
# diverse set of backbones in the initial model building step.  Allowing
# loophash to design the whole loop should help accomplish that.  We'll use
# LayerDesign in the next step to make sure we still end up with greasy
# residues on the inside and polar one on the outside.
#
# Positions 44+45: Design the active site loop up to G46.  We don't think it
# would be wise to include G46 in the design, since it seems to be responsible
# for breaking the α-helix at position 47.  Previously we stopped at position
# 43 in hopes of keeping its salt-bridge with E52, but Rosetta usually got rid
# of that interaction anyways.
#
# We'll be using LayerDesign and ConsensusLoopDesign to restrict which residues
# can go where, so the only thing we're specifying here is `NOTAA CH`.  We
# never want cysteine because it can mess up the global fold by forming
# unexpected disulfides, and we never want histidine because it's behavior can
# be pH dependent.

34      A NOTAA CH
35      A NOTAA CH
36      A NOTAA CH
37      A NOTAA CH
38      A PIKAA E
39      A NOTAA CH
40      A NOTAA CH
41      A NOTAA CH
42      A NOTAA CH
43      A NOTAA CH
44      A NOTAA CH
45      A NOTAA CH
198     B NOTAA CH
199     B NOTAA CH
200     B NOTAA CH
201     B NOTAA CH

# Repack positions
# ================
# The following repack positions were chosen by the clash-based repack
# shell creator (excluding the ligand).

14      A NATAA
30      A NATAA
49      A NATAA
50      A NATAA
53      A NATAA
54      A NATAA
94      A NATAA
108     A NATAA
110     A NATAA
```

```
111     A NATAA
112     A NATAA
113     A NATAA
114     A NATAA
115     A NATAA
120     A NATAA
126     B NATAA
203     B NATAA
224     B NATAA
226     B NATAA

# The following repack positions were added after visual inspection of
# clash-based repack shell.

10      A NATAA
13      A NATAA
17      A NATAA
25      A NATAA
51      A NATAA
52      A NATAA
55      A NATAA
56      A NATAA
57      A NATAA
107     A NATAA
109     A NATAA
116     A NATAA
117     A NATAA
125     B NATAA
127     B NATAA
227     B NATAA
```

Listing 2.15: Most recent loop file, for the input model with no deletions.

```
LOOP  26  51  40 0 0
LOOP 198 203 200 0 0
```

Listing 2.16: Most recent loop file, for the input model with one deletion.

```
LOOP  26  50  40 0 0
LOOP 197 202 199 0 0
```

2.6.2 Command lines

Below are the command lines that were used at each step of the PIP protocol. The following variables are placeholders for different input paths and parameters:

$ROSETTA Path to a Rosetta installation.

$ROSETTA_BUILD OS, compiler, and options used to build Rosetta, e.g. `linuxclangrelease`

$INPUT_PDB The input structure for the current step, e.g. an output structure from the previous step.

$NATIVE_PDB The structure of KSI D38E, with the native active site loop geometry.

$RESFILE The residues that are allowed to mutate.

$RESTRAINT The target coordinates for the E38 carboxylate.

$LOOP The region of backbone that was remodeled.

Listing 2.17: Command-line used for the build models step.

```
$ROSETTA/source/bin/loopmodel.$ROSETTA_BUILD             \
    -in:file:s $INPUT_PDB                                \
    -in:file:native $NATIVE_PDB                          \
    -in:file:extra_res_fa "EQU.fa.params"                \
    -in:file:extra_res_cen "EQU.cen.params"              \
    -in:file:fullatom                                    \
    -out:overwrite                                       \
    -out:pdb_gz                                          \
    -packing:ex1                                         \
    -packing:ex2                                         \
    -packing:extrachi_cutoff 0                           \
    -packing:resfile $RESFILE                            \
    -constraints:cst_fa_weight 1.0                       \
    -constraints:cst_fa_file $RESTRAINT                  \
    -loops:loop_file $LOOP                               \
    -loops:remodel "perturb_kic"                         \
    -loops:refine "refine_kic"                           \
    -loops:kic_rama2b                                    \
    -loops:kic_omega_sampling                            \
    -loops:allow_omega_move "true"                       \
    -loops:ramp_fa_rep                                   \
    -loops:ramp_rama                                     \
```

Listing 2.18: Command-line used for the design models step.

```
$ROSETTA/source/bin/fixbb.$ROSETTA_BUILD                 \
    -in:file:s $INPUT_PDB                                \
    -in:file:extra_res_fa "EQU.fa.params"                \
    -in:file:extra_res_cen "EQU.cen.params"              \
    -out:overwrite                                       \
    -out:pdb_gz                                          \
    -packing:ex1                                         \
    -packing:ex2                                         \
    -packing:extrachi_cutoff 0                           \
    -packing:use_input_sc                                \
    -packing:resfile $RESFILE                            \
```

Listing 2.19: Command-line used for the validate designs step.

```
$ROSETTA/source/bin/loopmodel.$ROSETTA_BUILD             \
    -in:file:s $INPUT_PDB                                \
    -in:file:native $NATIVE_PDB                          \
    -in:file:extra_res_fa "EQU.fa.params"                \
    -in:file:extra_res_cen "EQU.cen.params"              \
    -in:file:fullatom                                    \
    -out:pdb_gz                                         \
    -out:overwrite                                      \
    -packing:ex1                                        \
    -packing:ex2                                        \
    -packing:extrachi_cutoff 0                          \
    -loops:loop_file $LOOPS                             \
    -loops:remodel "perturb_kic"                        \
    -loops:refine "refine_kic"                          \
    -loops:kic_rama2b                                   \
    -loops:kic_omega_sampling                           \
    -loops:ramp_fa_rep                                  \
    -loops:ramp_rama                                    \
```

2.6.3 Manual screening

Design	# Del	Round	Number	Loop Sequence	Loop Cluster	COOH Offset (Å)	Score Gap (REU)	% Sub-Å Offset	pKa	Δ Buried Unsats
	0	2	49	AKWEELLSVPIYP	1	0.75	-5.97	2.80	10.02	14
	0	1	68	AKWEELLDVPIYP	1	0.98	-3.55	0.80	9.82	14
	0	2	33	AEFKEQIPGQVGR	2	1.10	-6.09	2.20	7.81	6
	0	1	35	ARFEEQIPGQVGQ	2	0.51	-4.79	0.60	9.10	7
	0	2	32	AEFVEQIPGQIGR	2	1.09	-3.20	5.20	9.92	7
	0	1	29	ARFEEQIPGMVGQ	2	1.15	-0.15	1.60	7.68	10
D	0	1	87	AIRRERYAKANPR	3	0.67	-2.91	1.20	5.74	8
	0	1	83	AWTYEYIGPPGGN	4	1.18	-1.63	0.40	9.13	3
	0	2	30	AEVTETKYPEPRR	5	0.69	-3.47	5.20	9.17	1
E	0	2	1	AKVIETQYPEPRK	5	0.68	-6.56	6.60	9.11	2
	0	2	5	AEVIETQYPEPRR	5	0.70	-1.71	8.00	9.34	2
	0	2	10	AEVTETKYPEPRR	5	0.78	-4.09	4.80	9.14	2
	0	2	29	AKVTETKYPLPMK	5	0.87	-5.82	3.80	8.63	2
G	0	2	9	ARVEETKYPEDRK	5	0.94	-13.76	0.80	8.76	2
	0	2	6	AEVIETQYPEPRR	5	0.66	-10.14	7.00	9.07	3
	0	2	8	AEVIETQYPYPKR	5	0.68	-8.51	8.82	8.92	3
	0	2	22	AKVTETMYPEPRK	5	0.71	-7.68	6.60	9.15	3
	0	2	17	AKVTETKYPEPRK	5	0.77	-7.89	7.00	8.94	3
	0	1	22	AKVTETKYPEDRK	5	0.78	-6.87	2.61	9.05	3
	0	1	17	AKVTETKYPEDRK	5	0.81	-3.35	6.01	9.03	3
	0	2	27	AEVTETKYPEPRR	5	0.91	-6.50	2.61	8.79	3
	0	2	16	ASVTETKYPEDRT	5	0.99	-8.92	5.60	8.81	3
F	0	2	4	AKVIETQYPEPRK	5	0.66	-11.96	7.41	8.96	4
	0	2	0	AEVRETQYPEDRR	5	0.72	-10.74	5.40	8.94	4
	0	2	7	AKVIETQYPEPRK	5	0.82	-6.95	8.40	8.88	4
	0	2	12	AKVTETKYPEPRK	5	0.93	-7.47	5.80	8.72	4
	0	2	39	AEVTETQYPTNFR	5	0.65	-1.27	10.60	8.95	5
	0	2	15	AKVRETKYPEPRK	5	0.97	-10.38	4.40	8.79	5
	0	2	25	AKVIETQYPYDFQ	5	0.61	-5.44	13.20	9.32	6
	0	2	3	ARVEETQYPEDRK	5	0.64	-10.55	4.21	8.96	6
	0	2	44	AKVTETKYPTDFK	5	0.65	-2.39	8.40	9.08	6
	0	2	41	AKVTETKYPTDFK	5	0.81	-3.07	13.43	8.75	6
	0	2	2	AEVIETQYPEPRR	5	1.05	-9.11	9.22	8.70	6
	0	2	46	ASVIETKYPNDYT	5	0.64	-5.35	9.20	9.15	7
	0	2	28	AKVIETQYPYDFK	5	0.71	-2.52	13.20	9.29	7
	0	2	42	AKVIETKYPNDYK	5	0.83	-5.20	11.00	8.76	8
	0	2	24	AKVIETKYPYDFQ	5	0.79	-9.21	16.83	8.74	9
	0	2	34	AKVIETKYPYDFQ	5	0.81	-6.42	16.80	8.80	9
	0	2	36	AKVIETKYPNDYK	5	0.77	-5.35	12.25	8.81	11
A	0	1	6	AYVEESAGQPKYW	6	1.05	-0.32	0.60	7.68	4
	0	1	54	ARVWEGGLTQWYK	7	1.09	-0.13	1.80	8.58	10
	0	1	59	AQVVESFYRGWPP	8	0.91	-3.82	12.60	8.54	11
	0	2	21	AFWYELTDYPWYP	9	0.87	-7.69	18.18	8.85	9
	0	2	31	AFWYELTDYPWYP	9	0.77	-7.70	15.20	9.07	10
	0	2	26	AFYYELTDKPWYP	9	0.95	-10.72	10.40	6.88	10
	0	2	14	AFWYENTDKPWYP	9	0.77	-0.34	10.04	8.91	11
	0	1	44	ANYTESQNPDIRG	10	1.04	-3.42	1.20	7.33	8
B	0	1	25	AKVTEDAGLGGYQ	11	0.86	-1.64	7.41	8.66	2
	0	1	38	GWLEEQYGYWKYS	12	1.17	-5.27	1.60	9.82	10
	0	1	81	GYRREQNGFWKYT	12	0.87	-2.88	1.00	6.37	12
	0	1	37	GRREEQFGWENYS	12	1.01	-7.26	1.20	6.27	14
	0	2	38	AEWNEQFGWRGNP	13	0.53	-2.28	1.40	9.14	9
	0	1	10	AIWNEQYGWRGRP	13	0.61	-3.21	1.20	8.85	9
C	0	1	47	ARRNEIGGPPPLP	14	0.92	-1.30	3.20	6.76	6
	0	1	74	ARYNEPYFDRDEK	15	0.73	-3.46	3.60	8.65	9
	0	1	73	AFYYEYNWGTWRP	16	0.89	-6.29	0.60	7.13	15
	0	2	47	AFYYESNSGDWYP	17	1.18	-0.23	1.00	6.81	10
	0	1	91	AELGEGEDTGIPR	18	1.13	-2.11	0.00	8.68	6
	0	1	56	ADSPEGGPWSSYR	19	0.71	-0.23	0.60	7.90	5

Design	# Del	Round	Number	Loop Sequence	Loop Cluster	COOH Offset (Å)	Score Gap (REU)	% Sub-Å Offset	pKa	Δ Buried Unsats
	1	1	82	GRADENGTGTYK	1	0.74	-0.99	2.20	7.83	12
	1	1	86	AQFDEGDQGWPN	2	0.96	-0.34	1.20	8.65	6
	1	1	60	GYRDELKYPPLP	3	0.89	-3.95	2.00	6.29	8
	1	1	61	ARYEEQYYPPKE	3	0.48	-1.37	1.81	7.71	9
	1	1	66	ARYEELYFPPLP	3	0.80	-0.85	3.40	7.41	9
	1	1	78	ARYEEQYFPPLS	3	0.46	-2.69	2.20	7.31	10
	1	2	46	AQYNEIGFPGGS	4	0.58	-1.95	12.42	7.82	6
N	1	2	44	AQYNEIGFRGDS	4	0.33	-4.30	17.80	7.75	7
	1	2	5	ARYIERGFPNQP	4	0.43	-2.88	20.64	7.43	7
	1	2	43	AKYIERGFPNLP	4	0.56	-0.55	18.67	7.33	7
	1	2	23	AQYVEEGFPNLP	4	0.95	-2.22	23.80	9.84	7
	1	2	38	AQYVERGFPNKP	4	0.30	-0.70	20.64	7.55	8
	1	1	67	AQYDEIGFRGDS	4	0.40	-4.78	12.40	7.30	8
	1	2	30	AQYNEIGFRGDS	4	0.38	-3.34	18.40	7.70	9
	1	2	11	AQYVERGFPNLP	4	0.50	-1.24	22.24	7.43	9
	1	2	34	AQYDEIGFRGDP	4	0.66	-2.48	25.90	7.27	9
	1	2	40	ARYVEEGFPNLP	4	0.86	-1.12	22.24	9.61	9
	1	2	20	AQYVERGFPNNP	4	0.30	-0.17	21.80	7.54	10
	1	2	14	AQYVERGFPNKP	4	0.38	-0.52	19.40	7.41	10
	1	2	35	ARYIERGFPNLP	4	0.40	-1.81	18.91	7.56	10
	1	2	15	AQYDEIGFRGDP	4	0.41	-1.82	22.74	7.78	10
	1	2	28	AQYVERGFPNLP	4	0.54	-0.62	23.20	8.80	10
	1	1	58	AQYDEIGFRGDS	4	0.64	-4.41	16.20	9.21	10
	1	2	6	ARYIERGFPNQP	4	0.27	-2.12	24.60	7.81	11
	1	2	24	AQYDEIGFRGDP	4	0.47	-3.31	26.60	8.89	11
	1	2	42	AQYVERGFPNKP	4	0.61	-0.26	22.85	7.90	11
	1	2	26	AQYVEIGFPNLP	4	0.86	-3.04	36.40	9.57	11
	1	2	41	ARYVERGFPNMP	4	0.31	-1.02	24.05	7.77	12
	1	1	49	AQYVEIGFPNLP	4	0.91	-1.28	39.68	9.29	12
	1	2	29	AQYFEIGFRGDP	4	0.41	-2.47	26.25	7.75	13
	1	2	37	AQYVEIGFPNLP	4	0.78	-2.22	47.49	9.38	13
	1	2	32	ARYVEIGFPNLP	4	0.82	-2.50	38.96	9.52	13
	1	2	25	AQYVEIGFPNLP	4	0.83	-2.48	37.80	9.63	13
I	1	1	28	ARYDEIGFPDTG	5	0.38	-2.84	6.60	7.88	5
	1	1	34	ADAEERGFPPLT	5	0.60	-1.19	0.80	7.48	5
	1	2	45	AWVYEQGYPIGP	6	0.81	-4.05	4.21	8.86	5
	1	1	53	AWVYEQGYPIGP	6	0.61	-3.59	3.80	9.35	6
L	1	2	7	ATVTESFRPPFT	7	1.06	-3.46	10.20	8.75	0
	1	2	0	ATVRESFRPPFT	7	1.12	-3.57	9.02	8.88	0
M	1	2	21	ASVTESFRPPFT	7	0.75	-2.93	16.03	8.86	1
	1	2	3	AEVRESFRPPFR	7	0.97	-2.24	12.40	8.92	1
	1	2	1	ATVRESFRPPFT	7	1.10	-4.19	8.05	8.73	1
K	1	2	4	ATVRESFRPPFT	7	0.81	-4.75	17.80	8.94	2
	1	1	13	ATVREAFRPPFT	7	0.75	-3.66	21.80	8.84	3
	1	1	19	ATVTEAFRPPFT	7	0.74	-3.06	31.80	8.72	4
	1	2	13	AAVRESFRPPFK	7	0.77	-4.61	19.20	8.71	4
H	1	1	5	AIRYEQYYEGGK	8	0.64	-3.97	2.81	6.11	4
	1	1	0	AERFEQYYEGGR	8	0.74	-4.19	1.60	6.47	4
	1	1	7	AIRYEQYYEGGK	8	0.66	-2.17	2.20	6.20	5
	1	1	11	ATRYEQYYEGGT	8	0.70	-6.68	0.80	6.16	5
J	1	1	62	AQYDEIGFDGGS	9	0.46	-2.01	7.40	7.21	2
	1	1	64	GYRDEQFGWKWT	10	0.83	-6.28	2.80	6.29	11
	1	1	56	ARVEEWLGYGGQ	11	1.12	-1.40	0.60	7.70	3

Table 2.6: Quality metrics for the designs that were manually screened. Name: The names assigned to the designs that were picked for experimental validation. # Del: The number of deletions. Round: The round of computational design and validation that produced the design. Number: The number assigned to each design within its round. Loop Sequence: The residue identities for the positions where the backbone was remodeled. Loop Cluster: Designs were clustered according to the C/Cα/N/O RMSD for the positions where the backbone was remodeled (Listings 2.6,2.7). Clusters were formed such that the RMSD between any two designs in the same cluster was no greater than 1.2Å. COOH Offset: The furthest distance between any of the atoms in the E38 carboxylate and their target positions, for the lowest scoring loop modeling decoy. Score Gap: The difference in score between the lowest scoring decoy that puts all of the atoms of the E38 carboxylate less that 1Å from their target positions, and the lowest scoring decoy that puts at least one atom of the E38 carboxylate more than 2Å from its target position. % Sub-Å Offset: The fraction of the loop modeling decoys that are predicted to position the all atoms of the Glu carboxylate less than 1Å from their target positions. pKa: The of E38 as predicted by PROPKA3.0 [17]. D38 in wildtype KSI has a pKa of 4.5 [1], but is predicted by PROPKA3.0 to be 6.2 (PDB: 8CHO). Δ Buried Unsats: The change in the number of buried unsatisfied H-bonds relative to wildtype KSI, as calculated by the BuriedUnsatFilter in Rosetta.

Design	# Dels	# Muts	# Revs	COOH Offset (Å)	Score Gap (REU)	% Sub-Å Offset	Picked	Reversions																				
A	0	26	0	1.05	-0.32	0.60																						
		10	16	2.33	0.00	1.00		Y30F		P32D	N33D	R48T	D49A	N50A	K53E	K57N		A60K		N75A	G76N	I109V	S110V		A112M		I115L	D118E
		16	10	1.05	-2.05	0.20	✓	Y30F		P32D		R48T		N50A		K57N		A60K		N75A	G76N	I109V	S110V					
B	0	29	0	0.86	-1.64	7.41																						
		12	17	0.83	-3.16	7.60	✓	Y30F	D31A		T33D	R48T	D49A		W53E	K57N	A58S	E60K		N75A	G76N	I109V	S110V	Y111S	A112M	Q113R		D118E
C	0	29	0	0.92	-1.30	3.21	✓																					
		14	15	1.58	-1.12	0.80			D31A		T33D	R48T	D49A	N50A		E57N		A60K	T74V	N75A	G76N	I109V	S110V	Y111S		Q113R		D118E
		19	10	1.02	-0.49	1.80		Y30F	D31A			R48T	D49A			E57N				N75A	G76N		S110V	Y111S				D118E
D	0	30	0	0.67	-2.91	1.20	✓																					
		14	16	1.45	-2.38	0.20		Y30F	S31A	P32D		R48T	D49A		Q53E	E57N		A60K	T74V	N75A	G76N	I109V	A110V		A112M		I115L	D118E
		18	12	2.30	0.00	0.40			S31A	P32D		R48T			Q53E			A60K	T74V	N75A	G76N	I109V	A110V		A112M		I115L	
		23	7	0.67	-6.08	0.40				P32D								A60K	T74V	N75A		I109V			A112M		I115L	
E	0	26	0	0.68	-6.56	6.60	✓																					
		8	18	0.61	-1.17	4.80		Y30F		P32D	N33D	R48T	D49A	N50A		E57N	A58S	R60K		N75A	G76N	I109V	N110V	Y111S	A112M	Q113R	V115L	P118E
		11	15	0.61	-5.37	4.80		Y30F		P32D		R48T	D49A	N50A		E57N		R60K		N75A	G76N	I109V	N110V	Y111S	A112M	Q113R		P118E
		16	10	0.83	-4.84	3.00						R48T	D49A	N50A				R60K		N75A		I109V		Y111S	A112M	Q113R		P118E
F	0	26	0	0.66	-11.96	7.41																						
		8	18	0.66	-4.51	3.40		Y30F		P32D	N33D	R48T	D49A	N50A		L57N	Q58S	A60K		N75A	G76N	I109V	N110V	Y111S	A112M	Q113R	V115L	P118E
		9	17	0.55	-6.50	7.20	✓	Y30F		P32D	N33D	R48T	D49A	N50A		L57N	Q58S	A60K		N75A	G76N	I109V	N110V	Y111S		Q113R	V115L	P118E
G	0	26	0	0.94	-13.76	0.80																						
		9	17	0.88	-1.77	2.00		Y30F		P32D	N33D	R48T	D49A	N50A		L57N	Q58S			N75A	G76N	I109V	S110V	Y111S	A112M	Q113R	V115L	D118E
		13	13	0.92	-12.21	2.20	✓	Y30F		P32D		R48T	D49A	N50A		L57N				N75A	G76N	I109V	S110V	Y111S			V115L	D118E
H	1	27	0	0.64	-3.97	2.81	✓																					
		10	17	0.54	-5.71	3.00		Y30F	D31A	S32D	T33D		D48A	N49A	K52E	M56N			T73V	N74A	G75N	I108V	A109V	E110S	A111M		I114L	D117E
		12	15	0.62	-2.93	2.00		Y30F	D31A	S32D			D48A	N49A	K52E	M56N			T73V	N74A	G75N	I108V	A109V	E110S			I114L	D117E
		20	7	0.67	-4.01	1.60		Y30F	D31A							M56N			T73V				A109V	E110S			I114L	
I	1	26	0	0.38	-2.84	6.60	✓																					
		19	7	0.88	-1.63	7.00				P32D			D48A			K56N			T73V	N74A							I114L	D117E
J	1	26	0	0.46	-2.01	7.40	✓																					
		19	7	2.08	0.00	1.80										K56N				N74A	G75N		A109V	E110S	A111M		I114L	
K	1	25	0	0.81	-4.75	17.80																						
		9	16	1.13	-0.70	3.01		Y30F		P32D	N33D		E48A	N49A	R52E	K56N	N57S			N74A	G75N	I108V	A109V	E110S	A111M		V114L	D117E
		12	13	0.83	-4.15	9.00		Y30F		P32D	N33D		E48A	N49A	R52E	K56N					G75N	I108V	A109V		A111M		V114L	D117E
		13	12	0.80	-5.09	11.40	✓	Y30F		P32D	N33D		E48A	N49A	R52E	K56N					G75N	I108V	A109V				V114L	D117E
L	1	26	0	1.06	-3.46	10.20																						
		8	18	0.93	-2.37	6.00		Y30F	D31A	S32D	T33D		E48A	N49A	R52E	K56N	A57S			N74A	G75N	I108V	S109V	Y110S	A111M	Q112R	V114L	D117E
		15	11	0.72	-2.21	6.00			D31A		T33D			N49A	R52E	K56N	A57S					I108V	S109V		A111M		V114L	D117E
		17	9	0.57	-4.72	11.20	✓		D31A		T33D			N49A	R52E	K56N	A57S						S109V				V114L	D117E
M	1	27	0	0.75	-2.93	16.03																						
		9	18	0.79	-0.38	4.20		Y30F	D31A	S32D	T33D		E48A	N49A	R52E	K56N	N57S			N74A	G75N	I108V	S109V	Y110S	A111M	Q112R	V114L	P117E
		12	15	0.87	-7.40	13.20	✓	Y30F	D31A	S32D	T33D		E48A	N49A	R52E	K56N					G75N	I108V	S109V	Y110S		Q112R	V114L	P117E
N	1	25	0	0.33	-4.30	17.80																						
		9	16	2.76	0.00	3.20		Y30F		P32D	N33D		D48A	N49A		R56N	Q57S		T73V	N74A	G75N	I108V	S109V	Y110S		Q112R	N114L	D117E
		13	12	0.80	-4.12	16.60	✓	Y30F		P32D	N33D		D48A	N49A		R56N			T73V			I108V	S109V			Q112R	N114L	D117E

Table 2.7: Wildtype reversions for each of the designs selected for experimental validation. Design: The name of the design. # Dels: The number of deletions. # Muts: The number of mutations relative to wildtype, excluding deletions. # Rev: The number of wildtype reversions. COOH Offset, Score Gap, % Sub-Å Offset: As described in Table 2.6. Picked: The set of reversions that was chosen for experimental testing. Reversions: The specific reversion mutations that were made for each design. The mutations are aligned by position (although the deletion causes the indices to differ in some columns).

2.6.4 Experimental validation

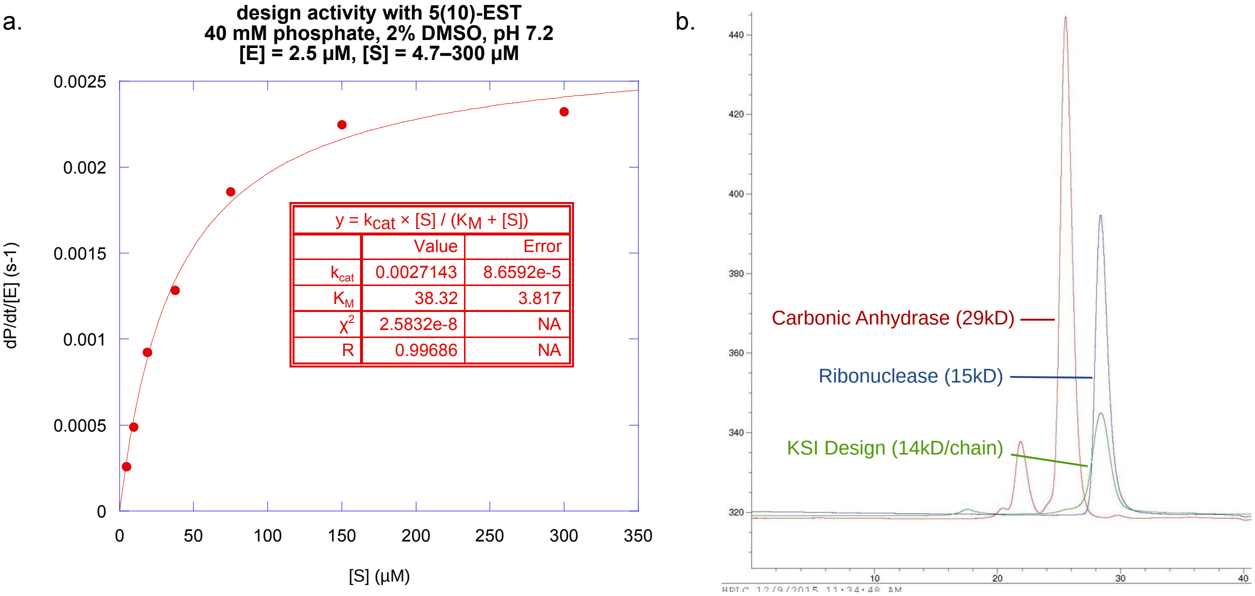

Figure 2.6: Experimental characterization of the most active KSI design. (a) Determination of k_{cat} and K_M. (b) Size exclusion chromatography traces, showing that the KSI design is a monomer in solution.

2.6.5 Sequences

Listing 2.20: Amino acid sequences for the designs that were experimentally validated.

```
>A
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADNAYVEESAGQPKYWGTDAIRKFYANQLKLPLAVELTQEVRAAANEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKVVSARAIFGDKNIHAGA
>B
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADDAKVTEDAGLGGYQGTAWIREFYANSLKLPLAVELTQEVRANANEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKVVSMRATFGEKNIHAGA
>C
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYDDTARRNEIGGPPPLPGRDNIRKFYAEDLALPLAVELTQEVRATNGEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKISYAQAVFGDKNIHAGA
>D
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYSPDAIRRERYAKANPRGRDAIRQFYAEDLALPLAVELTQEVRATNGEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKIAEAQAIFGDKNIHAGA
>E
```

```
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYAPNAKVIETQYPEPRKGRDNIREFYAEALRLPLAVELTQEVRAANGEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKINYAQAVFGPKNIHAGA
>F
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADDAKVIETQYPEPRKGTAAIREFYANSLKLPLAVELTQEVRAAANEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKVVSARALFGEKNIHAGA
>G
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADNARVEETKYPEDRKGTAAIREFYANQLALPLAVELTQEVRAAANEAA
FAFIVSFEYQGRKTVVAPIDHFRFNGAGKVVSAQALFGEKNIHAGA
>H
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYDSTAIRYEQYYEGGKGTDNIRKFYAMDLKLPLAVELTQEVRATNGEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKIAEARAIFGDKNIHAGA
>I
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYAPNARYDEIGFPDTGGTDNIRAFYAKQLKLPLAVELTQEVRATNGEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKIAEARAIFGDKNIHAGA
>J
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYDSTAQYDEIGFDGGSGTENIRRFYAKQLKLPLAVELTQEVRATNGEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKIAEARAIFGDKNIHAGA
>K
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADDATVRESFRPPFTGTAAIREFYANNLKLPLAVELTQEVRASNNEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKVVEAQALFGEKNIHAGA
>L
HMNTPEHMTAVVQRYVAALNAGDLDGIVALYASDATVTESFRPPFTGTEAIREFYANSLKLPLAVELTQEVRAANGEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKIVYAQALFGEKNIHAGA
>M
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADDASVTESFRPPFTGTAAIREFYANNLKLPLAVELTQEVRASNNEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKVVSARALFGEKNIHAGA
>N
HMNTPEHMTAVVQRYVAALNAGDLDGIVALFADDAQYNEIGFRGDSGTAAIREFYANQLKLPLAVELTQEVRAVNGEAAF
AFIVSFEYQGRKTVVAPIDHFRFNGAGKVVYARALFGEKNIHAGA
```

Listing 2.21: DNA sequences for the designs that were experimentally validated.

```
>A
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
```

```
CGTCGCGCTGTTTGCCGATAACGCCTATGTGGAAGAATCCGCGGGTCAGCCGAAATATTGGGGTACGGATGCGATTCGTA
AGTTTTACGCCAACCAGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGGCCGCCAACGAAGCGGCC
TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC
CGGCAAGGTGGTGAGCGCGCGCGCCATCTTTGGCGATAAGAATATTCACGCTGGCGCCTGAAGCTT
>B
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTTTGCCGATGACGCCAAGGTGACAGAAGACGCGGGTCTCGGGGGCTATCAGGGTACGGCTTGGATTCGTG
AGTTTTACGCCAACTCGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGAACGCCAACGAAGCGGCC
TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC
CGGCAAGGTGGTGAGCATGCGCGCCACGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT
>C
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTATGACGATACCGCCAGGAGGAATGAAATCGGGGGTCCCCCGCCCCTGCCCGGTAGGGATAATATTCGTA
AGTTTTACGCCGAAGATCTCGCACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGACCAACGGCGAAGCGGCC
TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC
CGGCAAGATCTCGTACGCGCAGGCCGTGTTTGGCGATAAGAATATTCACGCTGGCGCCTGAAGCTT
>D
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTATTCCCCTGACGCCATCCGGCGAGAACGCTATGCTAAGGCGAACCCGCGCGGTAGGGATGCGATTCGTC
AGTTTTACGCCGAAGATCTCGCACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGACCAACGGCGAAGCGGCC
TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC
CGGCAAGATCGCGGAGGCGCAGGCCATCTTTGGCGATAAGAATATTCACGCTGGCGCCTGAAGCTT
>E
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTATGCCCCTAACGCCAAGGTGATAGAAACCCAGTATCCCGAGCCCAGGAAGGGTAGGGATAACATTCGTG
AGTTTTACGCCGAGGCGCTCAGACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGGCCAACGGCGAAGCGGCC
TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC
CGGCAAGATCAATTACGCGCAGGCCGTGTTTGGCCCGAAGAATATTCACGCTGGCGCCTGAAGCTT
>F
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTTTGCCGATGACGCCAAGGTGATAGAAACCCAGTATCCCGAGCCCAGGAAGGGTACGGCTGCGATTCGTG
AGTTTTACGCCAACTCGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGGCCGCCAACGAAGCGGCC
TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC
```

```
CGGCAAGGTGGTGAGCGCGCGCGCCTTGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT

>G

CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT

CGTCGCGCTGTTTGCCGATAACGCCAGGGTGGAAGAAACCAAGTATCCCGAGGACAGGAAGGGTACGGCTGCGATTCGTG

AGTTTTACGCCAACCAGCTCGCACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGGCCGCCAACGAAGCGGCC

TTCGCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGC

CGGCAAGGTGGTGAGCGCGCAGGCCTTGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT

>H

CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT

CGTCGCGCTGTATGACAGTACCGCCATCCGGTATGAACAGTATTACGAGGGCGGGAAGGGTACGGATAATATTCGTAAGT

TTTACGCCATGGATCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGACCAACGGCGAAGCGGCCTTC

GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG

CAAGATCGCGGAGGCGCGCGCCATCTTTGGCGATAAGAATATTCACGCTGGCGCCTGAAGCTT

>I

CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT

CGTCGCGCTGTATGCCCCTAACGCCAGGTATGACGAAATCGGTTTCCCGGACACGGGCGGTACGGATAACATTCGTGCGT

TTTACGCCAAGCAGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGACCAACGGCGAAGCGGCCTTC

GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG

CAAGATCGCGGAGGCGCGCGCCATCTTTGGCGATAAGAATATTCACGCTGGCGCCTGAAGCTT

>J

CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT

CGTCGCGCTGTATGACAGTACCGCCCAGTATGACGAAATCGGTTTCGACGGCGGGTCCGGTACGGAAAATATTCGTCGGT

TTTACGCCAAGCAGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGACCAACGGCGAAGCGGCCTTC

GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG

CAAGATCGCGGAGGCGCGCGCCATCTTTGGCGATAAGAATATTCACGCTGGCGCCTGAAGCTT

>K

CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT

CGTCGCGCTGTTTGCCGATGACGCCACGGTGCGAGAATCGTTTCGCCCGCCCTTTACCGGTACGGCTGCGATTCGTGAGT

TTTACGCCAACAACCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGAGCAACAACGAAGCGGCCTTC

GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG

CAAGGTGGTGGAGGCGCAGGCCTTGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT

>L

CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
```

```
CGTCGCGCTGTATGCCAGTGACGCCACGGTGACAGAATCGTTTCGCCCGCCCTTTACCGGTACGGAAGCGATTCGTGAGT
TTTACGCCAACTCGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGGCCAACGGCGAAGCGGCCTTC
GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG
CAAGATCGTGTACGCGCAGGCCTTGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT
>M
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTTTGCCGATGACGCCTCGGTGACAGAATCGTTTCGCCCGCCCTTTACCGGTACGGCTGCGATTCGTGAGT
TTTACGCCAACAACCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGAGCAACAACGAAGCGGCCTTC
GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG
CAAGGTGGTGAGCGCGCGCGCCTTGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT
>N
CATATGAATACCCCAGAACACATGACCGCCGTGGTACAGCGCTATGTGGCTGCGCTCAATGCCGGCGATCTGGACGGCAT
CGTCGCGCTGTTTGCCGATGACGCCCAGTATAATGAAATCGGTTTCCGGGGCGACTCCGGTACGGCTGCGATTCGTGAGT
TTTACGCCAACCAGCTCAAACTGCCTTTGGCGGTGGAGCTGACGCAGGAGGTACGCGCGGTCAACGGCGAAGCGGCCTTC
GCTTTCATCGTCAGCTTCGAGTATCAGGGCCGCAAGACCGTGGTTGCGCCCATCGATCACTTTCGCTTCAATGGCGCCGG
CAAGGTGGTGTACGCGCGCGCCTTGTTTGGCGAGAAGAATATTCACGCTGGCGCCTGAAGCTT
```

2.7 References

[1] Ralph M. Pollack et al. “pH dependence of the kinetic parameters for 3-oxo-.DELTA.5-steroid isomerase. Substrate catalysis and inhibition by (3S)-spiro[5.alpha.-androstane-3,2’-oxiran]-17-one”. In: *Biochemistry* 25.8 (Apr. 1986), pp. 1905–1911. ISSN: 0006-2960. DOI: 10.1021/bi00356a011. URL: https://doi.org/10.1021/bi00356a011 (visited on 04/11/2018) (cit. on pp. 33, 86).

[2] Ralph M. Pollack, Joseph P. G. Mack, and Sherif Eldin. “Direct observation of a dienolate intermediate in the base-catalyzed isomerization of 5-androstene-3,17-dione to 4-androstene-3,17-dione”. In: *Journal of the American Chemical Society* 109.16 (Aug. 1987), pp. 5048–5050. ISSN: 0002-7863. DOI: 10.1021/ja00250a061. URL: https://doi.org/10.1021/ja00250a061 (visited on 05/15/2018) (cit. on p. 33).

[3] Jason P. Schwans et al. “Use of anion–aromatic interactions to position the general base in the ketosteroid isomerase active site”. en. In: *Proceedings of the National Academy of Sciences* 110.28 (July 2013), pp. 11308–11313. ISSN: 0027-8424, 1091-6490. DOI: 10.

1073/pnas.1206710110. URL: http://www.pnas.org/content/110/28/11308 (visited on 09/13/2016) (cit. on p. 33).

[4] Jason P. Schwans et al. “Experimental and Computational Mutagenesis To Investigate the Positioning of a General Base within an Enzyme Active Site”. In: *Biochemistry* 53.15 (Apr. 2014), pp. 2541–2555. ISSN: 0006-2960. DOI: 10.1021/bi401671t. URL: https://doi.org/10.1021/bi401671t (visited on 04/06/2018) (cit. on pp. 33, 34, 38, 40, 53).

[5] Daniel A. Kraut et al. “Dissecting the paradoxical effects of hydrogen bond mutations in the ketosteroid isomerase oxyanion hole”. en. In: *Proceedings of the National Academy of Sciences* 107.5 (Feb. 2010), pp. 1960–1965. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0911168107. URL: http://www.pnas.org/content/107/5/1960 (visited on 05/24/2018) (cit. on p. 34).

[6] Stephen D. Fried, Sayan Bagchi, and Steven G. Boxer. “Extreme electric fields power catalysis in the active site of ketosteroid isomerase”. en. In: *Science* 346.6216 (Dec. 2014), pp. 1510–1514. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1259802. URL: http://science.sciencemag.org/content/346/6216/1510 (visited on 05/24/2018) (cit. on p. 34).

[7] Daniel J. Mandell, Evangelos A. Coutsias, and Tanja Kortemme. “Sub-angstrom accuracy in protein loop reconstruction by robotics-inspired conformational sampling”. en. In: *Nature Methods* 6.8 (Aug. 2009), pp. 551–552. ISSN: 1548-7105. DOI: 10.1038/nmeth0809-551. URL: https://www.nature.com/articles/nmeth0809-551 (visited on 05/08/2018) (cit. on p. 37).

[8] Amelie Stein and Tanja Kortemme. “Improvements to Robotics-Inspired Conformational Sampling in Rosetta”. In: *PLOS ONE* 8.5 (May 2013), e63090. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0063090. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0063090 (visited on 11/18/2017) (cit. on pp. 37, 46, 47).

[9] Sarel J. Fleishman et al. “RosettaScripts: A Scripting Language Interface to the Rosetta Macromolecular Modeling Suite”. en. In: *PLOS ONE* 6.6 (June 2011), e20161. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0020161. URL: http://journals.plos.org/

plosone/article?id=10.1371/journal.pone.0020161 (visited on 05/24/2018) (cit. on p. 43).

[10] Andrew Leaver-Fay et al. “A Generic Program for Multistate Protein Design”. en. In: *PLOS ONE* 6.7 (July 2011), e20937. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0020937. URL: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0020937 (visited on 05/23/2018) (cit. on p. 43).

[11] Alexander M. Sevy et al. “Design of Protein Multi-specificity Using an Independent Sequence Search Reduces the Barrier to Low Energy Sequences”. en. In: *PLOS Computational Biology* 11.7 (July 2015), e1004300. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1004300. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004300 (visited on 05/23/2018) (cit. on p. 43).

[12] Patrick Löffler et al. “Rosetta:MSF: a modular framework for multi-state computational protein design”. en. In: *PLOS Computational Biology* 13.6 (June 2017), e1005600. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1005600. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005600 (visited on 05/23/2018) (cit. on p. 43).

[13] Michael D. Tyka et al. “Alternate states of proteins revealed by detailed energy landscape mapping”. In: *Journal of molecular biology* 405.2 (Jan. 2011), pp. 607–618. ISSN: 0022-2836. DOI: 10.1016/j.jmb.2010.11.008. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3046547/ (visited on 05/23/2018) (cit. on p. 44).

[14] Noah Ollikainen, René M. de Jong, and Tanja Kortemme. “Coupling Protein Side-Chain and Backbone Flexibility Improves the Re-design of Protein-Ligand Specificity”. en. In: *PLOS Computational Biology* 11.9 (Sept. 2015), e1004335. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1004335. URL: http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004335 (visited on 05/23/2018) (cit. on p. 44).

[15] Hyun-Soo Cho et al. “Crystal Structure and Enzyme Mechanism of Δ5-3-Ketosteroid Isomerase from Pseudomonas testosteroni,” in: *Biochemistry* 37.23 (June 1998), pp. 8325–8330. ISSN: 0006-2960. DOI: 10.1021/bi9801614. URL: https://doi.org/10.1021/bi9801614 (visited on 05/16/2018) (cit. on p. 46).

[16] Hyun-Soo Cho et al. “Crystal Structure of Δ5-3-Ketosteroid Isomerase from Pseudomonas testosteroni in Complex with Equilenin Settles the Correct Hydrogen Bonding Scheme for Transition State Stabilization”. en. In: *Journal of Biological Chemistry* 274.46 (Nov. 1999), pp. 32863–32868. ISSN: 0021-9258, 1083-351X. DOI: `10.1074/jbc.274.46.32863`. URL: `http://www.jbc.org/content/274/46/32863` (visited on 05/16/2018) (cit. on p. 46).

[17] Mats H. M. Olsson et al. “PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions”. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 525–537. ISSN: 1549-9618. DOI: `10.1021/ct100578z`. URL: `https://doi.org/10.1021/ct100578z` (visited on 04/11/2018) (cit. on p. 86).

Chapter 3

Controlling CRISPR-Cas9 with ligand-activated and ligand-deactivated sgRNAs

> This chapter is adapted from the bioRχiv preprint by Kale Kundert, James E Lucas, Kyle E Watters, Christof Fellmann, Andrew H Ng, Benjamin M Heineike, Christina M Fitzsimmons, Benjamin L Oakes, David F Savage, Hana El-Samad, Jennifer A Doudna, Tanja Kortemme entitled “Controlling CRISPR-Cas9 with ligand-activated and ligand-deactivated sgRNAs” It is contents are reproduced here under the Creative Commons Attribution (CC BY) License.

The CRISPR-Cas9 system provides the ability to edit, repress, activate, or mark any gene (or DNA element) by pairing of a programmable single guide RNA (sgRNA) with a complementary sequence on the DNA target. Here we present a new method for small-molecule control of CRISPR-Cas9 function through insertion of RNA aptamers into the sgRNA. We show that CRISPR-Cas9-based gene repression (CRISPRi) can be either activated or deactivated in a dose-dependent fashion over a >10-fold dynamic range in response to two different small-molecule ligands. Since our system acts directly on each target-specific sgRNA, it enables new applications that require differential and opposing temporal control of multiple genes.

3.1 Introduction

CRISPR-Cas9 has emerged as an immensely powerful system for engineering and studying biology due to its ability to target virtually any DNA sequence via complementary base pairing with a programmable single-guide RNA (sgRNA) [1]. This ability has been harnessed to edit genomes, repress [2] or activate [3, 4] gene expression, image DNA loci [5], generate targeted mutational diversity[6] and to modify epigenetic markers [7].

In addition to engineering CRISPR-Cas9 for diverse applications, there has also been broad interest in developing strategies to regulate CRISPR-Cas9 activity [8, 9]. Such strategies promise to mitigate off-target effects and allow the study of complex biological perturbations that require temporal or spatial resolution [9]. To date, most of the progress in this area has been focused on switching the activity of the Cas9 protein using chemical [10, 11, 12, 13, 14, 15, 16] or optical [17, 18, 19] inputs. A general issue with these approaches is that all target genes are regulated in the same manner, although this limitation can be addressed with orthogonal CRISPR-Cas9 systems [16, 20, 21].

An alternative but less explored strategy is to regulate the sgRNA instead of the Cas9 protein. Since the sgRNA is specific for each target sequence, controlling the sgRNA directly has the potential to independently regulate each target. This strategy has been approached using sgRNAs that sequester the 20 nucleotide target sequence (the spacer) only in the absence of an RNA-binding ligand [22, 23], ligand-dependent ribozymes that cause irreversible RNA cleavage [23, 24], ligand-dependent protein regulators recruited to the sgRNA to alter CRISPR function [12], and engineered antisense RNA to sequester and inactivate the sgRNA [25].

Here we describe a new method to engineer ligand-responsive sgRNAs by using RNA aptamers to directly affect functional interactions between the sgRNA, Cas9, and the DNA target (Figure 3.1a). In contrast to prior sgRNA-based methods, our approach can be used to both activate and deactivate CRISPR-Cas9 function in response to a small molecule. In addition, our approach requires only Cas9 and the designed sgRNAs. We further show that control of CRISPR-Cas9 function with our method is dose-dependent over a wide range of ligand concentrations and can be used to simultaneously execute different temporal programs for multiple genes within a single cell. We envision that this method will be broadly useful for regulating essentially all applications of CRISPR-Cas9-mediated biological engineering.

3.2 Results and Discussion

We sought to insert an aptamer into the sgRNA such that ligand binding to the aptamer would either activate or deactivate CRISPR-Cas9 function. We envisioned that ligand binding could either stabilize or destabilize a functional sgRNA conformation — bound to Cas9 and the DNA target — over other competing states in the ensemble (Figure 3.1a). We chose the theophylline aptamer [27] as a starting point because it is well-characterized and has high affinity for its ligand, which is cell permeable and is not produced endogenously.

We first asked which sites in the sgRNA were most responsive to the insertion of the theophylline aptamer and which strategies for linking the aptamer to the sgRNA were most effective. We designed aptamer insertions at each of the sgRNA stem loops at sites that are solvent-exposed in the Cas9/sgRNA/DNA ternary complex and exhibit various levels of tolerance to mutation [26]. These insertion sites are denoted the upper stem, nexus, and hairpin (Figure 3.1b). We tested three linking strategies aimed at stabilizing a functional sgRNA conformation in the presence of the ligand: (i) replacing parts of each stem with the aptamer, (ii) splitting the sgRNA in half and using the aptamer to bring the halves together, and (iii) designing strand displacements (i.e. sequences that allow for alternative base pairing in the apo and holo states) (Figure 3.1c, Table 3.1).

To test the resulting 86 designed sgRNAs, we used an *in vitro* assay to measure differential Cas9-mediated DNA cleavage in the presence and absence of theophylline. We identified theophylline-responsive sgRNAs for all three insertion sites (Figure 3.1d), with the most successful designs derived from the strand displacement linking strategy (Table 3.1). We confirmed that the activity of our designs depended on the concentration of theophylline, as would be expected if the ligand affects function through binding the aptamer-containing designed sgRNA (Figure 3.1e). In total, 10 designs were responsive to theophylline *in vitro*. For nine of these responsive designs theophylline addition activated CRISPR-Cas9 function, while for one design (#61) theophylline unexpectedly deactivated function. Interestingly, all of the theophylline-activated designs had the aptamer inserted into either the upper stem or the hairpin, while the theophylline-deactivated design had the aptamer inserted into the nexus (Figure 3.1b,d, Table 3.1). These findings suggested

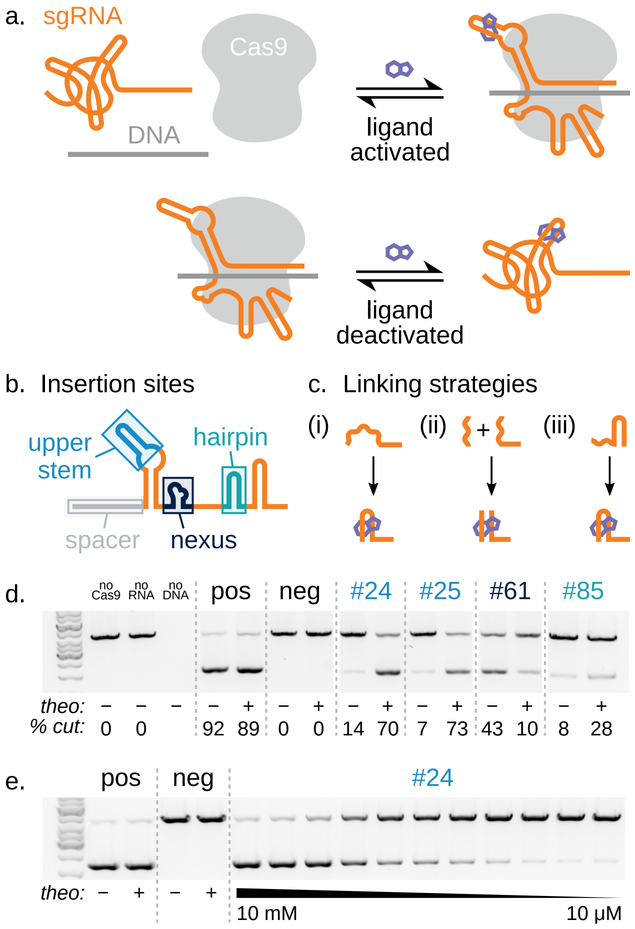

Figure 3.1: Design of ligand-controlled sgRNAs by inserting small molecule aptamers into the sgRNA. (a) Illustration of the design goal, where functional Cas9/sgRNA/target DNA complexes are stabilized either in the presence (top) or absence (bottom) of a small molecule ligand. (b) Aptamer insertion sites; sgRNA domains defined as in ref. [26]. (c) Strategies for linking the aptamer to the sgRNA: (i) stem replacement; (ii) induced dimerization; (iii) strand displacement (Table 3.1). (d) Efficiency of *in vitro* Cas9 cleavage of DNA in the presence and absence of 10 mM theophylline for controls and selected designs. Design numbers refer to Table 3.1 and are color-coded by aptamer insertion sites defined in (b). Percent cut values (bottom) are the average of at least two experiments. All data shown are from a single gel (some lanes are excluded for clarity). (e) Dose-dependence of cleavage efficiency in response to increasing concentrations of theophylline for a representative design (#24).

the exciting possibility of regulating CRISPR-Cas9 function with both ligand-activated and ligand-deactivated sgRNAs, depending on the aptamer insertion site.

We next sought to find designed sgRNAs that would function robustly in E. coli. To screen designed libraries using fluorescence-activated cell sorting (FACS), we changed to a cellular assay based on CRISPR-Cas9-mediated repression (CRISPRi) of super-folder green fluorescent protein (sfGFP) and monomeric red fluorescent protein (mRFP) (Figure 3.2a) [14]. The strongest rational designs exhibited only weak activity in the CRISPRi assay (Figure 3.4). Since the sequences linking the aptamer to the remainder of the sgRNA affected activity in our *in vitro* experiments, we designed libraries with randomized linkers of 4–12 nucleotides at all three insertion sites to broadly sample different aptamer contexts (Figure 3.2b, Table 3.2). To identify ligand-activated sgRNAs, we screened each library first for CRISPRi activity in the presence of theophylline, then second for lack of activity in the absence of theophylline. We then repeated that selection/counter-selection with a different spacer to avoid selecting sgRNA scaffold sequences that would be specific for a particular spacer (Figure 3.2c). To identify ligand-inhibited sgRNAs, we used an analogous four-step selection/counter-selection protocol that began by screening for activity in the absence of ligand. We validated the activity of the selected hits with a third spacer that was not used in any of the screens (Table 3.3). The most robust ligand-activated sgRNA variant (termed ligRNA$^+$; i.e. sgRNA that is active in the + ligand state) and ligand-inactivated sgRNA (termed ligRNA$^-$) showed 11x and 13x dynamic ranges that spanned 55% and 59% of the range achieved by the controls, with negligible overlap between the active and inactive populations (Figure 3.2d,e, Table 3.4).

The ligRNA$^+$ construct derived from inserting the aptamer into the hairpin while randomizing the remainder of the hairpin, the 5 unpaired nucleotides at the apex of the nexus, and the region between the hairpin and the nexus. The hairpin became more GC-rich, but the base-pairing was conserved with the exception of a single mismatch. The apex of the nexus became complementary to the 5’ region of the aptamer. The region between the hairpin and the nexus remained AU-rich and unpaired (Figure 3.5a). Secondary structure predictions of ligRNA$^+$ using ViennaRNA [29] (Figure 3.5b) are consistent with our intended mechanism, where ligand binding to the aptamer leads to strand displacements stabilizing the active sgRNA conformation.

The ligRNA$^-$ construct derived from inserting the aptamer into the nexus while randomizing the nexus stem. The nexus stem was extended from 2 to 5 bp, but remained base-paired and

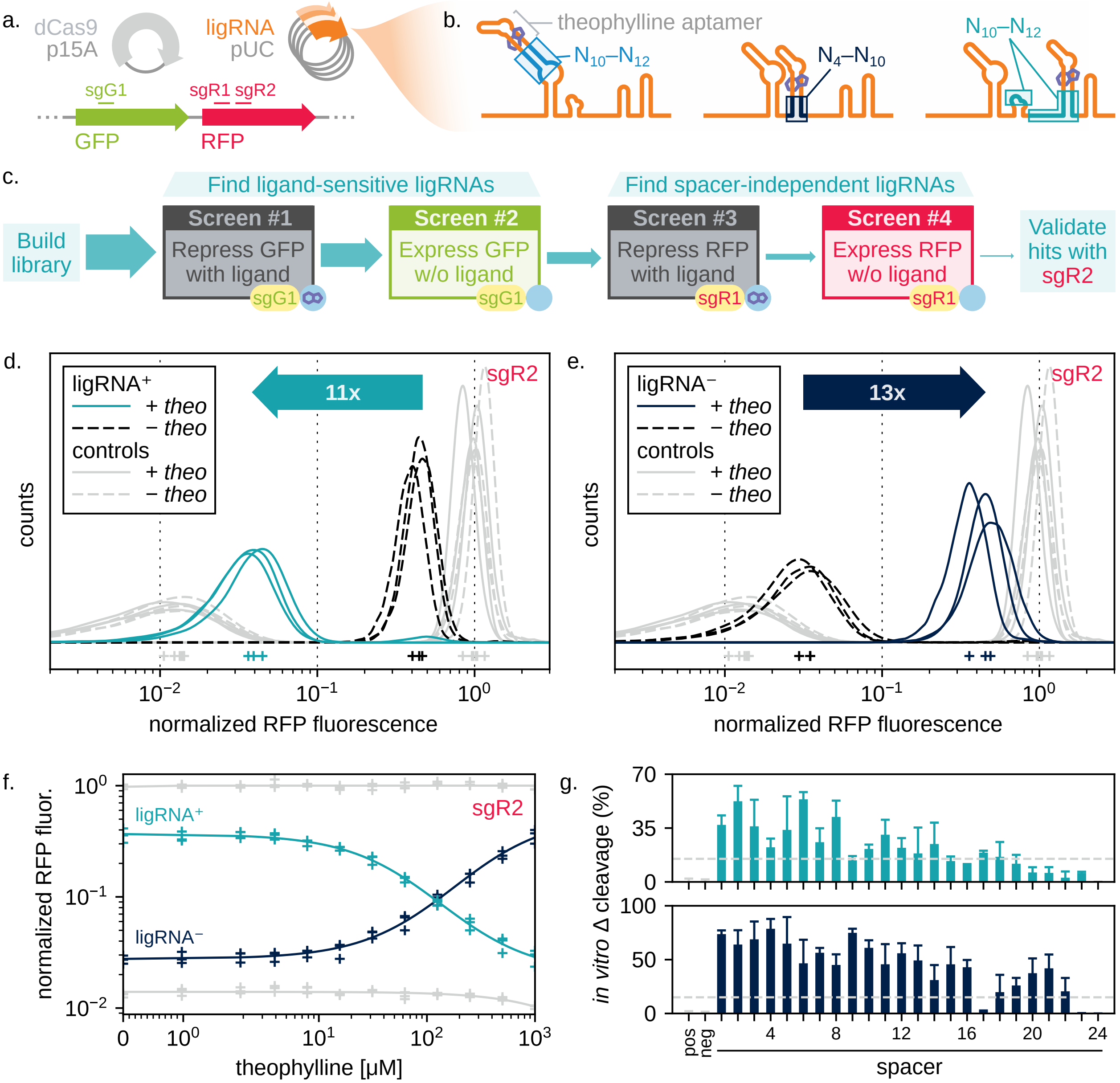

Figure 3.2: (caption on next page)

Figure 3.2: Identification of robust ligRNAs using CRISPRi-based gene repression in E. coli. (a) Components used in the CRISPRi assay. dCas9 and any ligRNAs were expressed from plasmids, while the fluorescent reporters (GFP and RFP) were chromosomally integrated. The DNA regions targeted by different spacers (sgG1, sgR1, sgR2) used to repress the fluorescent reporters are indicated. (b) Regions randomized in each ligRNA library. (c) Schematic of the screen used to isolate ligRNA$^+$. sgG1, sgR1, and sgR2 refer to spacers targeting GFP and RFP, respectively (Table 3.4). (d,e) Single-cell RFP fluorescence distributions for ligRNA$^+$ (teal, panel d) and ligRNA$^-$ (navy, panel e) targeting RFP using the sgR2 spacer with (solid lines) and without (dashed lines) theophylline. Control distributions are in grey (positive control: optimized sgRNA scaffold [28]; negative control: G43C G44C [26]). The mode of each distribution is indicated with a plus sign. RFP fluorescence values for each cell are normalized by both GFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. (f) Efficiency of CRISPRi repression with increasing theophylline concentrations for ligRNA$^+$ (teal) and ligRNA$^-$ (navy). Controls are in grey. The fluorescence axis is the same as in (d) and (e). The fits are to a two-state equilibrium model. (g) Change in the percentage of DNA cleaved *in vitro* in the presence and absence of theophylline for ligRNA$^+$ and ligRNA$^-$ in the context of 24 representative spacers. Each bar represents the mean of three or four measurements with a single spacer (except for spacer #24, where n=2, Table 3.5), and the error bars give the standard deviation. Pos and neg denote the positive and negative controls (Table 3.4), which are independent of theophylline.

GC-rich (Figure 3.5a). The mechanism underlying the ability of ligRNA$^-$ to deactivate CRISPR-Cas9 function upon the addition of theophylline was unclear. ViennaRNA predictions of the lowest energy conformation for ligRNA$^-$ were uninformative on the mechanism of ligand control, as they suggested that ligRNA$^-$ adopts the same secondary structure in the presence and absence of theophylline (Figure 3.5c). However, we noticed that in the stem sequence selected in our screen, U95 in ligRNA$^-$ (U59 in the crystal structure of the Cas9 ternary complex with DNA and RNA [30]) was conserved in 17 of the 20 isolated sequences (Table 3.3). This uracil makes specific hydrogen-bonding interactions with asparagine 77 in the ternary complex. In the sgRNA scaffold this uracil is always unpaired, but in ligRNA$^-$ it is predicted to engage in a wobble base pair with G65 in the stem leading up to the aptamer (Figure 3.5c). These observations led us to hypothesize that ligand binding to the aptamer controls the extent to which U95 is unpaired, which in turn determines whether or not ligRNA$^-$ interacts functionally with Cas9 and the target DNA. To test this hypothesis, we first designed strand-swapping mutations in the stem leading up to the aptamer (Figure 3.6). As expected, swapping U95 rendered ligRNA$^-$ completely inactive, while swapping base pairs at the positions between U95 and the aptamer had only a mild effect (Figure 3.6d). We then modulated the strength of the base pairs between U95 and the aptamer. Consistent with the hypothesis that ligand binding to the aptamer decreases access to U95, we found that weaker base pairs were more repressing while stronger base pairs were more activating (Figure 3.6e). These results provide a possible explanation for how ligRNA$^-$ deactivates CRISPR-Cas9 function in the presence of the ligand and suggest additional ways for ligRNAs to be tuned for specific applications.

Next, we tested whether the ligRNAs responded to increasing concentrations of theophylline in a dose-dependent manner in the cellular CRISPRi assay. We observed that the activities of both ligRNA$^+$ and ligRNA$^-$ were smoothly titratable and exhibited a nearly linear response over a large range of ligand concentration (Figure 3.2f). We note that the apparent EC50s of ligRNA$^+$ and ligRNA$^-$ (134.3±11.3 µM and 177.6±17.3 µM) are much higher than the KD of the theophylline aptamer alone (320 nM) [31]. This discrepancy is common for RNA devices [32] and could be explained by the altered structural context of the aptamer embedded in an sgRNA sequence. Nevertheless, the linear concentration dependence of the ligRNAs demonstrates their utility for not only turning genes on or off, but also for precisely tuning their levels of expression.

Because RNA devices are known to be sensitive to sequence context [33], we tested ligRNA$^+$ and ligRNA$^-$ with 24 different spacers using the *in vitro* DNA cleavage assay (Figure 3.2g, Figure 3.7, Table 3.5). We found that both ligRNA$^+$ and ligRNA$^-$ respond to theophylline for the majority of the tested spacers (15 and 21 out of 24 spacers for ligRNA$^+$ and ligRNA$^-$, respectively). For the few spacers that did not function, we hypothesized that base-pairing of the spacer sequence with the aptamer might explain the lack of sensitivity to theophylline. To address this question, we predicted the affinity between each spacer and the aptamer (with its associated linker) for both ligRNAs using ViennaRNA [29] (Figure 3.8). For ligRNA$^+$ constructs the correlation between the duplex free energy prediction and theophylline sensitivity was negligible. However, for ligRNA$^-$ constructs increased predicted affinity of the spacer for the aptamer sequence correlated with a smaller change in Cas9-mediated DNA cleavage in response to theophylline. This analysis suggested that spacers with predicted affinity for the aptamer could interfere with switching of the ligRNA$^-$ function, providing a useful design criterion for functional spacers. Taken together, these results suggest that ligRNAs should be capable of regulating most genes, especially those that can be targeted by multiple spacers.

To test how decreased expression levels would affect the ligRNAs, we replaced the strong constitutive promoter driving sgRNA expression (J23119) with a weak constitutive promoter (J23150)

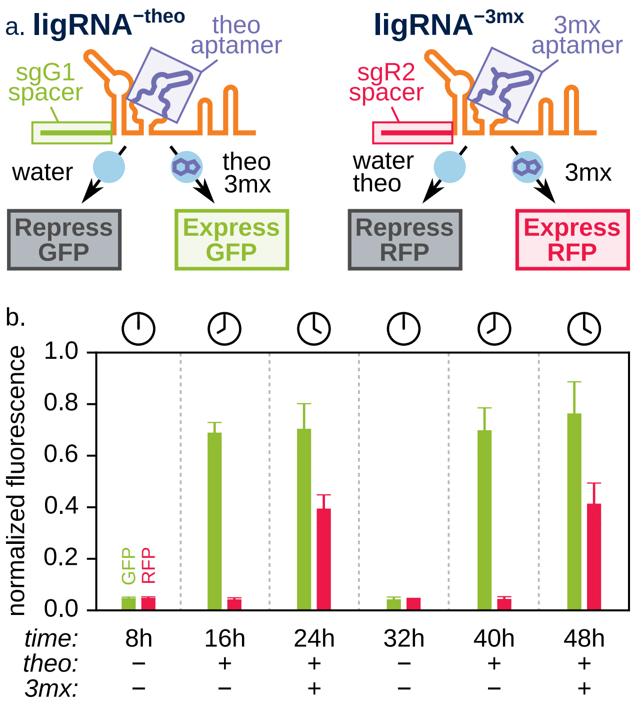

Figure 3.3: Multiplexed temporal control of two genes with two ligands. (a) Schematic illustrating the constructs and the expected consequences of adding theophylline (theo) and 3-methylxanthine (3mx) for each fluorescent reporter. (b) GFP and RFP fluorescence measured at indicated time points by flow cytometry. Presence of theo leads to GFP expression; addition of 3mx separately at a different time point also triggers RFP expression. Both effects are reversible (GFP and RFP are repressed when both ligands are absent) and expression can be triggered a second time with ligand addition. Bar heights and error bars represent the modes and standard deviations of the cell distributions, respectively. Unlike in Figure 3.2, fluorescence is normalized by side-scatter because both fluorescent channels are being manipulated.

and repeated the CRISPRi assay. Both ligRNAs remained functional with the weak promoter, albeit with a somewhat narrower dynamic ranges (from a 10.2±0.7-fold to a 5.9±0.5-fold change upon theophylline addition for ligRNA$^+$ and from a 16.2±1.0-fold to a 11.6±0.9-fold change for ligRNA$^-$, Figure 3.9). Notably, the weak promoter shifted the dynamic ranges of both ligRNAs in the direction of increased gene expression, to the point where nearly full gene activation was achieved in the non-repressing state. These results suggest that the ligRNAs are able to repress at low expression levels, and that tuning promoter strength is useful for applications that require full gene activation (alternatively, a collection of ligRNA variants that shift the dynamic range is shown in Figure 3.10).

A key advantage of regulating CRISPR-Cas9 using the sgRNA instead of the protein is the ability to independently control different genes with different ligands in the same Cas9 system. To test this idea, we replaced the theophylline (theo) aptamer in the ligRNAs with the 3-methylxanthine

(3mx) aptamer (the resulting sgRNA constructs were termed ligRNA$^{\pm 3mx}$). While the theophylline aptamer is recognized by both ligands, the 3-methylxanthine aptamer is specific to its ligand [34]. Since the aptamers differ in only one position, the replacement of the theophylline aptamer with the 3-methylxanthine aptamer was straightforward and led to a 3-methylxanthine-sensitive ligRNA^{-} variant without further optimization. (ligRNA^{+} also remained functional with the 3-methylxanthine aptamer but exhibited an undesirable albeit small ~2-fold response to theophylline, Figure 3.11). We used the two ligRNA^{-} variants to construct a system that expresses GFP upon addition of theophylline and expresses GFP and RFP when both ligands are added (Figure 3.3a). We then performed a timecourse where we sequentially activated, deactivated, and reactivated both reporter genes using ligRNAs and observed the expected temporal expression program (Figure 3.3b).

Taken together, ligRNAs provide control of both gene repression and gene activation and can be multiplexed for differential control of genes in the same system (Figure 3.3). While ligRNAs function robustly in bacteria, transferring them to eukaryotic systems will require further optimization (Figure 3.12, Figure 3.13) which could be achieved using methods similar to our optimization (Figure 3.2) of the initial rational designs (Figure 3.1). Nevertheless, there are already many useful applications for ligRNAs in bacteria. For example, many species of bacteria do not have facile genetic controls available, and ligRNAs provide such controls with a minimal footprint. Moreover, temporally controlled gene expression programs are thought to be important for key biological processes in bacteria [35], and ligRNAs provide a way to conduct large-scale screens to probe these programs and their role in the interactions between bacteria and their environments [36].

The study of subtle effects in complex biological systems will increasingly require the ability not just to probe individual genes, or to knock down different sets of genes, but to tune the expression of many different genes with fine temporal precision. ligRNAs provide this capability by adding ligand- and dose-dependent control of individual sgRNAs to the already powerful CRISPR-Cas9 technology.

3.3 Future work

3.3.1 Orthogonal ligands

We attempted to create ligRNAs that were responsive to a number of small-molecules other than theophylline and 3-methylxanthine. We were most successful with thiamine pyrophospate (TPP). Although TPP is a natural metabolite, its aptamer has been used to regulate access to ribosome binding sites (RBS) in E. coli grown in minimal media with or without thiamine [37]. It's notable that this system is responsive to small quantities of ligand: half-maximal and maximal response are seen at 0.6 µM and 2 µM, respectively. The $K_D$of the TPP aptamer is 0.1 µM [38], and although the total concentration of TPP in E. coli is as high as 3.9 µM [39], the concentration of free TPP (in the absence of exogenous thiamine) is presumably much lower.

To test the ability of the TPP aptamer to regulate sgRNA function, we created a variant of ligRNA$^-$ containing the TPP aptamer with no further optimization. It exhibited a 1.5x response to 500 µM thiamine in our CRISPRi assay (Figure 3.14). Although this response was slight, it was enough to justify screening an entire library of TPP ligRNAs via the same four-screen process we used to find theophylline-sensitive ligRNAs. We ultimately found 11 unique TPP-activated ligRNAs, each with a roughly 5x response to TPP (Figure 3.15, Table 3.3). We did not pursue these ligRNAs any further, but future work could focus on improving their response to ligand, screening for TPP-deactivated ligRNAs, and testing the TPP ligRNAs with different spacers.

We also attempted to create ligRNAs that were responsive to adenine, guanine, and ammeline. These three molecules are closely related: adenine and guanine are both purines, ammeline is similarly an aromatic heterocycle, and the corresponding aptamers differ in only 3 positions [40]. However, we were not able to create responsive ligRNAs for any of these molecules. We abandoned adenine because it was toxic at concentrations as low as 8 µM (Figure 3.16). This toxicity results from a depletion of the cell's GTP pool, due to there being an excess of adenine relative to guanine [41]. We abandoned guanine because it was insoluble in water and DMSO. It is soluble to 5 mM in 20 mM NaOH, but precipitates when diluted into media to 500 µM. We were able to screen for ammeline-responsive ligRNAs, but we were not able to find any. We attribute this failure to the low affinity (K_D= 1.2 µM) of the ammeline aptamer for its ligand. This low affinity is the result of how the ammeline aptamer was created: the adenine aptamer was mutated in 3 positions such that it had no detectable binding to adenine, then a panel of 80 heterocyclic compounds was screened to find those that bound the new aptamer, of which ammeline bound the strongest [40]. No further optimization was performed to improve affinity.

A possible future direction for this project would be to screen for fluoride-sensitive ligRNAs. The fluoride aptamer was suggested to us by Kyle Watters, our collaborator with expertise in RNA devices. Although the fluoride aptamer has low affinity (60 μM) for its ligand [42, 43], this is balanced by the ability to grow cells in high concentrations of fluoride (up to 1 mM) without causing toxicity [44].

3.3.2 Eukaryotic ligRNAs

None of the ligRNAs screened in bacteria were immediately transferable to either human (Figure 3.12) or yeast cells (Figure 3.13). To create ligRNAs were functional in such eukaryotic systems, we sought to screen for ligRNAs in those same systems. We focused on yeast since it was the more experimentally tractable option, and we hypothesized that ligRNAs identified in yeast could be transferable to other eukaryotic systems. However, it may also have been possible to conduct such screens in human cells, as sgRNA libraries with about 3×10^5 variants (1x coverage) can be transfected [45, 46, 47].

We created and screened theophylline-based libraries 36–45 (Table 3.2) as described in the Methods, but found no ligand-sensitive sgRNAs. The most likely explanation for this failure was that our library coverage was low because our transformation efficiency was poor. This could have been a consequence of our libraries being chromosomally integrated. Compared to being on a plasmid, this makes sgRNA expression more homogeneous across the population, but it may also impair transformation. Improving transformation efficiency and repeating this screen would be a promising direction for future work.

3.4 Methods

3.4.1 Constructs

All experiments used Cas9 from S. pyogenes (called Cas9 throughout), with mutations D10A and H840A for CRISPRi experiments (dCas9). Sequences of relevant ligRNAs, aptamers, spacers and controls are listed in Table 3.4.

3.4.2 *in vitro* DNA cleavage assay

sgRNA *in vitro* transcription: Linear, double-stranded template DNA was acquired either by ordering gBlocks® Gene Fragments from IDT (experiments in Figure 3.1d,e) or by cloning the desired sequence into a pUC vector and digesting it with EcoRI and HindIII (experiments in Figure 3.2g). Each construct contained a T7 promoter and a spacer that began with at least 3 Gs (Table 3.4). DNA template (10–50 ng) was transcribed using the HiScribe™ T7 High Yield RNA Synthesis Kit (NEB E2040S) and unincorporated ribonucleotides were removed with Zymo RNA Clean & Concentrator™-25 spin columns (Zymo R1018).

Target DNA: Target DNA was prepared using inverse PCR to clone the appropriate sequence into a modified pCR2.1 vector approximately 2.1 kb downstream of its XmnI site. The vector was then digested with XmnI (NEB R0194S) as follows: mix 43.5 µL ≈500 ng/µL miniprepped pCR2.1 DNA, 5.0 µL 10X CutSmart buffer, and 1.5 µL 20 U/µL XmnI; incubate at 37°C until no uncleaved plasmid is detectable on a 1% agarose gel (usually 30–60 min); dilute to 30 nM in 10 mM Tris-Cl, pH 8.5; store at -20°C.

Cas9 reaction: We adapted the following protocol from ref. [26]: mix 5.0 µL water or 30 mM theophylline (in water) and 1.5 µL 1.5 µM sgRNA (in water); incubate at 95°C for 3 min, then at 4°C for 1 min; prepare Cas9 master mix for 40 reactions: 241.0 µL water, 66.0 µL 10x Cas9 buffer (NEB B0386A), and 1.0 µL 20 µM Cas9 (NEB M0386T); add 7.0 µL Cas9 master mix; incubate at room temperature for 10 min; add 1.5 µL 30 nM target DNA; pipet to mix; incubate at 37°C for 1 h; prepare quenching master mix: 4.68 µL 20 mg/mL RNase A (Sigma R6148), 4.68 µL 20 mg/mL Proteinase K (Denville CB3210-5), and 146.64 µL 6x Orange G loading dye via master mix; add 3 µL quenching master mix; incubate at 37°C for 20 min, then at 55°C for 20 min; run the entire reaction (18 µL) on a 1% agarose/TAE/GelRed gel at 4.5 V/cm for 70 min.

Gel quantification: Band intensities were quantified using Fiji (1.51r) [48]. The background was subtracted from each image using a 50 pixels rolling ball radius. The fraction of DNA cleaved in each lane (f) was calculated as follows ($\text{pixels}_{2\text{kb}}$ and $\text{pixels}_{4\text{kb}}$ are the intensities of the cleaved and uncleaved bands, respectively):

$$f = \frac{\text{pixels}_{2\text{kb}}}{\text{pixels}_{4\text{kb}} + \text{pixels}_{2\text{kb}}}$$

The change in cleavage due to ligand (Δf) was calculated as follows (fapo and fholo are the fractions of DNA cleaved in the reactions without and with theophylline, respectively):

$$\Delta f = f_{\text{theo}} - f_{\text{apo}}$$

3.4.3 CRISPR-Cas9-based repression (CRISPRi) assay in E. coli

Strain: The strain used for all CRISPRi experiments was E. coli MG1655 with dCas9 (containing the D10A and H840A mutations) and ChlorR on a p15A plasmid (pgRNA-bacteria, Addgene 44251), sgRNA and AmpR on a pUC plasmid (pdCas9-bacteria, Addgene 44249), and sfGFP [49], mRFP [50], and KanR chromosomally integrated at the nsfA locus. This strain was originally described in ref. [2].

Flow cytometry: Overnight cultures of the CRISPRi strain above were inoculated from freshly picked colonies in 1 mL Lysogeny Broth (LB) medium with 100 µg/mL carbenicillin (100 mg/mL stock in 50% EtOH) and 35 µg/mL chloramphenicol (35 mg/mL stock in EtOH). The next morning, fresh cultures were inoculated in 15 mL culture tubes or 24-well blocks by transferring 4 µL of overnight culture into 1 mL EZ Rich Defined Medium (Teknova M2105) with 0.1% glucose, 1 µg/mL anhydrotetracycline, 100 µg/mL carbenicillin, 35 µg/mL chloramphenicol, with or without 1 mM theophylline (added from a 30mM stock dissolved in water). These cultures were then grown for 8h at 37°C with shaking at 225 rpm before GFP (488 nm laser, 530/30 filter) and RFP (561 nm laser, 610/10 filter) fluorescence were measured using a BD LSRII flow cytometer. Approximately 10,000 events were recorded for each measurement. Biological replicates were performed on different days using different colonies from the same transformation.

Data analysis: Cell distributions were obtained by computing a Gaussian kernel density estimation (KDE) over the base-10 logarithms of the measured fluorescence values. The mode was considered to be the center of each distribution (e.g. for determining fold changes) and was obtained through the Broyden-Fletcher-Goldfarb-Shanno (BFGS) maximization of the KDE. Dose response curves were fit to the Hill equation (*y* is the normalized fluorescence, *x* in the theophylline concentration, EC50 is the inflection point, and y_{min} and y_{max} are the lower and upper asymptotes of the fit):

$$y = y_{\text{min}} + \frac{y_{\text{max}} - y_{\text{min}}}{1 + \text{EC50}/x}$$

3.4.4 Identification of functional ligRNAs in E. coli using FACS screens

Library generation: Randomized regions were inserted into the sgRNA using inverse polymerase chain reaction (PCR) with phosphate-modified and high-performance liquid chromatography (HPLC)-purified primers containing degenerate nucleotides.

Electrotransformation: Electrocompetent cells were prepared as follows: make ”low-salt” Super Optimal Broth (SOB) medium: 20 g bacto-tryptone, 5 g bacto-yeast extract, 2 mL 5 M NaCl, 833.3 µL 3 M KCl, water to 1L, pH to 7.0 with NaOH, autoclave 30 min at 121°C; pick a fresh colony and grow overnight in 1 mL SOB; in the morning, inoculate 1 L SOB with the entire overnight culture; grow at 37°C with shaking at 225 rpm until OD=0.4 (≈4 h); place cells in an ice bath for 10 min; wash with 400 mL pre-chilled water, then 200 mL pre-chilled water, then 200 mL pre-chilled 10% glycerol; resuspend in a total volume of 6 mL pre-chilled 10% glycerol; make 100 µL aliquots; flash-freeze and store at -80°C. Electrocompetent cells were transformed as follows: thaw competent cells on ice for 10 min; pipet once to mix cells with 2 µL ≈250 ng/µL library plasmid; shock at 1.8 kV with a 5 ms decay time; immediately add 1 mL pre-warmed SOB with catabolite repression (SOC) medium; recover at 37°C for 1 h; dilute into selective liquid media and grow at 37°C with shaking at 225 rpm overnight. After PCR and ligation, libraries were first transformed into electrocompetent Top10 cells, then mira-prepped [51], sequenced, and combined to achieve approximately equal representation of variants based on library size and DNA concentration, then transformed again into electrocompetent MG1655 cells already harboring the dCas9 plasmid.

Cell sorting: Cells were grown as for the CRISPRi assay, but when starting new cultures, care was taken to subculture at least 10x more cells than the size of the library (often 200 μL). Sorting was done using a BD FACSAria II cell sorter. Sorting was no slower than 1000 evt/s and no faster than 20,000 evt/s, with the slower speeds being more accurate and the faster speeds being necessary to sort large libraries. Gates were drawn based on the position of the control population if possible, and based on the most extreme library members otherwise. Typically the gates included between 1% and 5% of the population being sorted. All gates were drawn diagonally in GFP vs. RFP space. Sorted cells were collected in 1 mL SOC at room temperature and, after sorting, were diluted into selective media and grown at 37°C with shaking at 225 rpm overnight.

Screening for ligRNA$^+$: Pool libraries 23–28 from Table 3.2. First screen: grow without ligand, gate for GFP expression, sort 10,000 evt/s for 3.5 h. Second screen: grow with ligand, gate for GFP repression, sort 1500 evt/s for 70 min. Third screen: grow without ligand, gate for GFP expression, sort 1700 evt/s for 10 min. Fourth screen: grow with ligand, gate for GFP repression, sort 1000 evt/s for 2 min. Fifth screen: grow without ligand, gate for GFP expression, sort 5000 evt. Plate cells and test 96 individual colonies using the CRISPRi assay. Miniprep and sequence the 20 selected designs with the largest response to theophylline. Only one unique sequence was identified, and it did not function with the sgR1 spacer (Table 3.3). Note that we did not change the spacer in between the second and third screens for this library. We then designed libraries 29–30 (Table 3.2) to keep the stem identified in the previous screen of libraries 23-28 and to randomize other regions of the sgRNA that might be participating in ligand-dependent base-pairing. First screen: grow with ligand, gate for GFP repression, sort 4000 evt/s for 2 h. Second screen: grow without ligand, gate for GFP expression, sort 1500 evt/s for 1 h. Change the spacer from sgG1 to sgR1 (Table 3.7) using inverse PCR. Third screen: grow with ligand, gate for RFP repression, sort 2000 evt/s for 10 min. Fourth screen: grow without ligand, gate for RFP expression, sort 10,000 evt. Plate cells and test 96 individual colonies using the CRISPRi assay. Miniprep and sequence the 15 selected designs with the largest response to theophylline, then test those designs with four different spacers (sgG1, sgR1, sgG2, sgR2). There was only 1 duplicate sequence, and 8 of the sequences had acquired unexpected mutations outside of the randomized region. The majority of these hits were tested with four different spacers (sgG1, sgR1, sgG2, and sgR2). ligRNA$^+$ and ligRNA$^-$ performed best (none of the hits performed well with the sgG2 spacer.) Details on all library hits and validation with different spacers can be found in Table 3.3.

Screening for ligRNA⁻: Pool libraries 7–22 from Table 3.2. First screen: grow without ligand, gate for GFP repression, sort 18,000 evt/s for 1 h. Second screen: grow with ligand, gate for GRP expression, sort 1000 evt/s for 10 min. Third screen: grow without ligand, gate for GFP repression, sort 1000 evt/s for 10 min. Fourth screen: grow with ligand, gate for GRP expression, sort 1000 evt/s for 10 min. Fifth screen: grow without ligand, gate for GFP repression, sort 1500 evt/s for 7 min. Plate cells and test 96 individual colonies using the CRISPRi assay described above. Miniprep and sequence the 20 selected designs with the largest fold response to theophylline. In this group there were only 9 unique sequences. ligRNA⁻ appeared 5 times and had the largest response to theophylline (Table 3.3). Note ligRNA⁻ is functional with other spacers (Figure 3.2g) despite the fact that we did not change the spacer in between the second and third screens for this library.

3.4.5 Test of different spacers

The spacers for this assay were chosen by a script that generated uniformly random sequences, scored them using a previously published machine learning approach for designing functional sgRNAs [52], and kept only those that scored higher than 0.5 (the median). This approach was designed to produce spacers that were as unbiased as possible, while still being likely to function as expected in the positive (optimized sgRNA scaffold [28]) and negative (G43C G44C [26]) controls shown in Table 3.4. The average cleavage for the positive controls was 93%, and the lowest cleavage for any of the positive controls was 78% (Table 3.5).

3.4.6 RNA secondary structure predictions

Secondary structure predictions were performed using the `RNAfold` program from the ViennaRNA package (version 2.4.3). The structures reported here are minimum free energy (MFE) predictions, although centroid and maximum expected accuracy (MEA) structures from partition function calculations were nearly identical in every case. The holo state was simulated using soft constraints: a -9.21 kcal/mol bonus was granted for forming the base pair flanking the aptamer. This bonus corresponds to the 320 nM affinity of the theophylline aptamer for its ligand [31]. The command-lines used for the apo and holo states, respectively, are given below:

```
$ RNAfold --partfunc --MEA
$ RNAfold --partfunc --MEA --motif \
    "GAUACCAGCCGAAAGGCCCUUGGCAGC,(...((((((....)))...)))...),-9.212741321099747"
```

Free energy predictions for Figure 3.5 were performed using the RNAduplex program from the ViennaRNA package (version 2.4.3):

```
$ RNAduplex
```

3.4.7 CRISPRi assay in S. cerevisiae

Yeast Strains: All yeast strains were constructed using the MoClo golden gate cloning framework and the Yeast Toolkit from [53]. The background strain was WCD230 (derived from BY4741 (MATa his3Δ1 leu2Δ0 met15Δ0 ura3Δ0) with a larger fraction of the his3 gene removed [54]. All yeast strains contained the dCas9-Mxi1 inhibitor, fluorescent reporters, and ligRNA or control sgRNA construct, respectively (Figure 3.13a). The inhibitor cassette was integrated into the Ura3 locus and contained pGal1-dCas9-Mxi1 and pRnr2-GEM constructs that allowed for estradiol-inducible expression of dCas9-Mxi1 [55]. The Mxi1 inhibition domain was derived from Addgene catalog number 46921 [3] with synonymous mutations made to the E68 and T69 codons to render the construct compatible with golden gate cloning. The fluorescent reporter cassette was integrated into the His3 locus and contained pCcw12-sfGFP and pTdh3-mRFP. The guide cassette was integrated into the Leu2 locus and contained (tRNAPhe)-HDV Ribozyme-sgRNA. The HDV Ribozyme cleaved the ligRNA or control sgRNA from the rest of the transcript to prevent unwanted interactions [56]. The sgRNA was either positive or negative control sgRNA, ligRNA$^+$, ligRNA$^+_2$, or ligRNA$^-$.

Experimental protocol: Strains were plated from freezer stocks on SD-HIS (Yeast Nitrogen Base (YNB), Complete Synthetic Media (CSM) lacking histidine, 2% glucose) because the fluorescent reporter cassette integrated into the His3 locus caused a slight growth defect that was also present for an empty vector integrated in the same locus. After incubating at 30°C, single colonies were grown overnight in 0.5 mL YPD (yeast extract, peptone, 2% glucose) in 96 well plates with 2 mL/well maximum capacity, shaking at 900 RPM at 30°C in an Infors HT Multitron Pro shaker. Saturated cultures were diluted 1:100 into 1 mL SDC (YNB, CSM, 2% glucose) in a new 2 mL 96 well plate and placed at 30°C shaking at 900RPM for 2 hours. Cells were then diluted 1:4 into 400 µL SDC with estradiol and either theophylline from a 30 mM stock or water in a new 2 mL 96 well plate. The final concentration of estradiol was 125 nM and the final concentration of theophylline when present was 2.5 mM. After 8 hours shaking at 900 rpm and 30°C, cells were diluted in 1X TE buffer and analyzed on a flow cytometer (BD LSR II).

3.4.8 Gene editing assay in mammalian cells

HEK293T (293FT; Thermo Fisher Scientific) cells, and derived cell lines, were grown in Dulbecco's Modified Eagle Medium (DMEM; Corning Cellgro, #10-013-CV) supplemented with 10% fetal bovine serum (FBS; Seradigm #1500-500), and 100 Units/mL penicillin and 100 µg/mL streptomycin (Pen-Strep; Life Technologies Gibco, #15140-122) at 37°C with 5% CO2. HEK293T and HEK-RT1 cells were tested for absence of mycoplasma contamination (UC Berkeley Cell Culture facility) by fluorescence microscopy of methanol fixed and Hoechst 33258 (Polysciences #09460) stained samples.

HEK293T-based genome editing reporter cells, referred to as HEK-RT1, were established in a two-step procedure. In the first step, puromycin resistant monoclonal HEK-RT3-4 reporter cells were generated as previously described (Hui Liu et al., in revision). In brief, HEK293T human embryonic kidney cells were transduced at low-copy with the amphotropic pseudotyped retrovirus RT3GEPIR-sh.Ren.713 [57], comprising an all-in-one Tet-On system enabling doxycycline-controlled EGFP expression. After puromycin (2.0 µg/ml) selection of transduced HEK239Ts, 36 clones were isolated and individually assessed for i) growth characteristics, ii) homogeneous morphology, iii) sharp fluorescence peaks of doxycycline (1 µg/ml) inducible EGFP expression, iv) relatively low fluorescence intensity to favor clones with single-copy reporter integration, and v) high transfectability. HEK-RT3-4 cells are derived from the clone that performed best in these tests. In the second step, HEK-RT1 cells were derived by transient transfection of HEK-RT3-4 cells with vectors encoding Cas9 and sgRNAs targeting puromycin, followed by identification of monoclonal reporter cell lines that are puromycin sensitive.

A lentiviral vector, referred to as pCF204, expressing a U6 driven sgRNA and an EFS driven Cas9-P2A-Puro cassette was based on the lenti-CRISPR-V2 plasmid [58], by replacing the sgRNA with an enhanced Streptococcus pyogenes Cas9 sgRNA scaffold [5]. All sgRNAs (sgRen71: TAGGAATTATAATGCTTATC, sgGFP1: CCTCGAACTTCACCTCGGCG, sgGFP9: CCGGCAAGCTGCCCGTGCCC) were designed with a G preceding the 20-nt guide for better expression, and cloned into the lentiviral vector using the BsmBI restriction sites. The lentiviral vectors expressing ligRNA$^+$, ligRNA2+ and ligRNA$^-$ (referred to as pCF441, pCF442 and pCF443, respectively) were all based on pCF204, by replacing the SpyCas9 sgRNA scaffold with the respective ligRNAs using custom oligonucleotides (IDT), gBlocks (IDT), standard cloning methods, and Gibson assembly techniques. Lentiviral particles were produced using HEK293T packaging cells; viral supernatants were filtered (0.45µm) and added to target cells. Transduced HEK-RT1 target cells were selected on puromycin (1.0 µg/ml).

GFP expression in HEK-RT1 reporter cells was induced using doxycycline (1 µg/ml; Sigma-Aldrich). Percentages of GFP-positive cells were assessed by flow cytometry (Attune NxT, Thermo Fisher Scientific), routinely acquiring 10,000-30,000 events per sample. Theophylline (Sigma-Aldrich, #T1633-50G) was used at the indicated concentrations, ranging from 0.1 mM to 10 mM. Note, theophylline concentrations of 5 mM and 10 mM resulted in considerable cellular toxicity in the HEK293T-based reporter cell line.

3.4.9 Screen for different ligands

TPP media: 1x MOPS (Teknova M2101), 1x K2HPO4 (Teknova M2102), 1x Supplement EZ (Teknova M2104), 0.4% glucose (Teknova G0520), 0.2% casamino acids, 500 µM thiamine.

Adenine/Guanine media: 1x MOPS (Teknova M2101), 1x K2HPO4 (Teknova M2102), 1x ACGU (Teknova M2103), 0.4% glucose (Teknova G0520), 8–2000 µM adenine/guanine.

3.4.10 FACS screen in S. cerevisiae

Libraries 36–45 were prepared as described for the bacterial screens, then were transformed into yeast by the method of Gietz and Schiestl [59]. The order of the screens and the gating strategies were as described for the bacterial screens. In between the second and third screen, the spacer was swapped via colony PCR and Golden Gate cloning. Colony PCR: mix 46.1 µL water, 3.3 µL OD≈10 library culture, 0.6 µL 5 U/µL zymolase; incubate for 30 min at 37°C, then for 10 min at

95°C; PCR amplify the sgRNA library using the zymolase reaction as the template and primers containing the second spacer sequence and BsaI sites on either side of the sgRNA. Golden Gate cloning: 1.0 µL Yeast Toolkit [53] destination vector, 25.0 µL PCR product, 3.1 µL 10 T4 ligase buffer (NEB), 1.0 µL T4 ligase (NEB), 1.0 µL BsaI-HF (NEB); incubate for 30 cycles of 42°C and 16°C for 5 min each; electrotransform into Top10 cells as described previously.

A different strategy was used to change the spacer in libraries 36–38. Since the randomized region in these libraries was located just 6 bp from the spacer, there was no room for an inverse PCR primer to bind without overlapping the spacer. We instead designed primers that overlapped the 6 bp before the randomized region plus 10 bp of the spacer, followed immediately by a BtgZI restriction site. BtgZI cleaves 10 bp away from where it recognizes, and thus removes the spacer information from the amplified sgRNA. The amplified sgRNA and a destination vector including the desired spacer preceding a complementary BtgZI site were separately digested, gel purified, ligated, and electrotransformed into Top10 cells.

3.5 Appendix

#	Strategy	Domain	Cleavage (%)				N	Active?	Sequence
			apo	*holo*	Δ	σ			
1	Positive Control		87	87	0	5	13		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
2	Negative Control		0	0	-0	0	13		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAACCCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
3	Stem Replacement	Upper Stem	32	47	15	15	4		GGGGCCACUAGGACAGGAUGUUUUAGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
4	Stem Replacement	Upper Stem	100	100	0		1		GGGGCCACUAGGACAGGAUGUUUUAGA----U-------------AUACCAGCCGAAAGGCCCUUGGCAG-------------U----AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
5	Stem Replacement	Upper Stem	66	81	15		2	✓	GGGGCCACUAGGACAGGAUGUUUUAGA----UU------------AUACCAGCCGAAAGGCCCUUGGCAG------------UU----AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
6	Stem Replacement	Upper Stem	40	46	6	5	3		GGGGCCACUAGGACAGGAUGUUUUAGA----UUU-----------AUACCAGCCGAAAGGCCCUUGGCAG-----------UUU----AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
7	Stem Replacement	Upper Stem	61	79	18		2	✓	GGGGCCACUAGGACAGGAUGUUUUAGAG-----------------AUACCAGCCGAAAGGCCCUUGGCAG-----------------CAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
8	Stem Replacement	Upper Stem	76	86	10		2		GGGGCCACUAGGACAGGAUGUUUUAGAG---U-------------AUACCAGCCGAAAGGCCCUUGGCAG-------------U---CAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
9	Stem Replacement	Upper Stem	75	78	3		2		GGGGCCACUAGGACAGGAUGUUUUAGAG---UU------------AUACCAGCCGAAAGGCCCUUGGCAG------------UU---CAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
10	Stem Replacement	Upper Stem	63	66	3		2		GGGGCCACUAGGACAGGAUGUUUUAGAG---UUU-----------AUACCAGCCGAAAGGCCCUUGGCAG-----------UUU---CAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
11	Stem Replacement	Upper Stem	84	83	-0		2		GGGGCCACUAGGACAGGAUGUUUUAGAGC--UU------------AUACCAGCCGAAAGGCCCUUGGCAG------------UU--GCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
12	Stem Replacement	Upper Stem	51	53	1		2		GGGGCCACUAGGACAGGAUGUUUUAGAGC--UUUCCC--------AUACCAGCCGAAAGGCCCUUGGCAG--------UUUCCC--GCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
13	Stem Replacement	Upper Stem	63	66	4		2		GGGGCCACUAGGACAGGAUGUUUUAGAGC--UUUCCCUUUC----AUACCAGCCGAAAGGCCCUUGGCAG----UUUCCCUUUC--GCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
14	Stem Replacement	Upper Stem	36	31	-6		1		GGGGCCACUAGGACAGGAUGUUUUAGAGC--UUUCCCUUUCCCUUAUACCAGCCGAAAGGCCCUUGGCAGUUUCCCUUUCCCUU--GCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
15	Stem Replacement	Upper Stem	97	95	-2		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAUU------------AUACCAGCCGAAAGGCCCUUGGCAG------------UUUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
16	Stem Replacement	Upper Stem	72	72	-0		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAUUUCCC--------AUACCAGCCGAAAGGCCCUUGGCAG--------UUUCCCUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
17	Stem Replacement	Upper Stem	96	97	1		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAUUUCCCUUUC----AUACCAGCCGAAAGGCCCUUGGCAG----UUUCCCUUUCUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
18	Stem Replacement	Upper Stem	93	87	-5		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAUUUCCCUUUCCCUUAUACCAGCCGAAAGGCCCUUGGCAGUUUCCCUUUCCCUUUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
19	Stem Replacement	Lower Stem	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUU---------------------AUACCAGCCGAAAGGCCCUUGGCAG-----------------------AAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
20	Stem Replacement	Lower Stem	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUA--------------------AUACCAGCCGAAAGGCCCUUGGCAG----------------------UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
21	Stem Replacement	Lower Stem	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUA------UU------------AUACCAGCCGAAAGGCCCUUGGCAG------------UU--------UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
22	Stem Replacement	Lower Stem	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUA------UUUC----------AUACCAGCCGAAAGGCCCUUGGCAG----------UUUC--------UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
23	Stem Replacement	Lower Stem	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUA------UUUCCC--------AUACCAGCCGAAAGGCCCUUGGCAG--------UUUCCC--------UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
24	Strand displacement	Upper Stem	14	70	56	19	4	✓	GGGGCCACUAGGACAGGAUGUUUUA----------------ACUUAUACCAGCCGAAAGGCCCUUGGCAGAA-GU-------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
25	Strand displacement	Upper Stem	7	73	67	7	3	✓	GGGGCCACUAGGACAGGAUGUUUUA----------------ACUUAUACCAGCCGAAAGGCCCUUGGCAGAG-GU-------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
26	Strand displacement	Upper Stem	2	16	13		2		GGGGCCACUAGGACAGGAUGUUUUA----------------ACUUAUACCAGCCGAAAGGCCCUUGGCAGAC-GU-------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
27	Strand displacement	Upper Stem	6	35	29		2	✓	GGGGCCACUAGGACAGGAUGUUUUA----------------ACUUAUACCAGCCGAAAGGCCCUUGGCAGAAAGU-------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
28	Strand displacement	Upper Stem	1	1	1		1		GGGGCCACUAGGACAGGAUGUUUUA----------------ACUUAUACCAGCCGAAAGGCCCUUGGCAGGG-GU-------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
29	Strand displacement	Lower Stem	75	75	-0	3	3		GGGGCCACUAGGACAGGAUAGCCUUGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUAAGGCUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
30	Strand displacement	Lower Stem	76	81	5		2		GGGGCCACUAGGACAGGAUAGCCUUGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUAGGGCUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
31	Strand displacement	Lower Stem	2	5	3		1		GGGGCCACUAGGACAGGAUAGCCUUGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUAACGCUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
32	Strand displacement	Lower Stem	0	0	0		1		GGGGCCACUAGGACAGGAUAGCCUUGA------------------AUACCAGCCGAAAGGCCCUUGGCAG-----------------AAGUAAGAGCUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
33	Strand displacement	Upper Stem	74	77	3	7	3		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGU------AUACCAGCCGAAAGGCCCUUGGCAG------ACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
34	Strand displacement	Upper Stem	80	80	-0	0	3		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUUAA---AUACCAGCCGAAAGGCCCUUGGCAG---UUAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
35	Strand displacement	Upper Stem	47	52	6	6	3		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUUAAAAUAUACCAGCCGAAAGGCCCUUGGCAGAUUUUAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
36	Strand displacement	Upper Stem	91	94	3		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGAU-----AUACCAGCCGAAAGGCCCUUGGCAG------ACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
37	Strand displacement	Upper Stem	94	94	-0		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUU-----AUACCAGCCGAAAGGCCCUUGGCAG-----AACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
38	Strand displacement	Upper Stem	95	95	-0		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGCU-----AUACCAGCCGAAAGGCCCUUGGCAG-----AACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
39	Strand displacement	Upper Stem	94	95	1		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGAUU----AUACCAGCCGAAAGGCCCUUGGCAG-----AACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
40	Strand displacement	Upper Stem	94	94	-0		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUUA----AUACCAGCCGAAAGGCCCUUGGCAG----UAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
41	Strand displacement	Upper Stem	94	94	0		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUUG----AUACCAGCCGAAAGGCCCUUGGCAG----UAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
42	Strand displacement	Upper Stem	95	94	-1		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGCUA----AUACCAGCCGAAAGGCCCUUGGCAG----UAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
43	Strand displacement	Upper Stem	95	95	-0		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUAUA---AUACCAGCCGAAAGGCCCUUGGCAG----UAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
44	Strand displacement	Upper Stem	46	57	11		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGCUAA---AUACCAGCCGAAAGGCCCUUGGCAG---UUAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
45	Strand displacement	Upper Stem	93	93	0		2		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUUAAA--AUACCAGCCGAAAGGCCCUUGGCAG--UUUAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
46	Strand displacement	Upper Stem	14	38	24		2	✓	GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGCUAAA--AUACCAGCCGAAAGGCCCUUGGCAG--UUUAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
47	Strand displacement	Upper Stem	67	79	13		1		GGGGCCACUAGGACAGGAUGUUUUAGA--------UCGUUGAA--AUACCAGCCGAAAGGCCCUUGGCAG--UUUAACGA--------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
48	Strand displacement	Lower Stem	0	0	0	0	3		GGGGCCACUAGGACAGGAUUCGGCUGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUAGCCGAAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
49	Strand displacement	Lower Stem	70	72	2	3	3		GGGGCCACUAGGACAGGAUGUUAUCGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUGAUAACAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
50	Strand displacement	Lower Stem	65	69	4		2		GGGGCCACUAGGACAGGAUGUUGUCGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUGAUAACAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
51	Strand displacement	Lower Stem	27	32	4		2		GGGGCCACUAGGACAGGAUGGUAUCGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUGAUACCAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
52	Strand displacement	Lower Stem	9	11	2		1		GGGGCCACUAGGACAGGAUGCUAUCGA------------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUGAUAACAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
53	Strand displacement	Lower Stem	8	13	5		1		GGGGCCACUAGGACAGGAUGUUAAUCGA-----------------AUACCAGCCGAAAGGCCCUUGGCAG------------------AAGUGAUAACAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
54	Induced Dimerization	Upper Stem	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUAGA------------------AUACCAGCC GGCCCUUGGCAG------------------AAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
55	Induced Dimerization	Upper Stem	0	0	0		2		GGGGCCACUAGGACAGGAUGUUUUAGAG-----------------AUACCAGCC GGCCCUUGGCAG-----------------CAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
56	Induced Dimerization	Upper Stem	0	0	0		2		GGGGCCACUAGGACAGGAUGUUUUAGAGC----------------AUACCAGCC GGCCCUUGGCAG----------------GCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
57	Induced Dimerization	Upper Stem	0	0	0		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCU---------------AUACCAGCC GGCCCUUGGCAG---------------AGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
58	Induced Dimerization	Upper Stem	0	0	0		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUA--------------AUACCAGCC GGCCCUUGGCAG--------------UAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
59	Stem Replacement	Nexus	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAA-----AUACCAGCCGAAAGGCCCUUGGCAG-----GUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
60	Stem Replacement	Nexus	0	0	0		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAG----AUACCAGCCGAAAGGCCCUUGGCAG----CGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
61	Stem Replacement	Nexus	43	10	-32	8	5	✓	GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGG---AUACCAGCCGAAAGGCCCUUGGCAG---CCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
62	Stem Replacement	Nexus	66	60	-6		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGG---AUACCAGCCGAAAGGCCCUUGGCAG--UCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
63	Stem Replacement	Nexus	5	4	-1		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGC--AUACCAGCCGAAAGGCCCUUGGCAG-GUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
64	Stem Replacement	Nexus	95	98	3		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGU--AUACCAGCCGAAAGGCCCUUGGCAG--UCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
65	Stem Replacement	Nexus	84	70	-14		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGUU-AUACCAGCCGAAAGGCCCUUGGCAG-UUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
66	Stem Replacement	Nexus	34	26	-8		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGUUUAUACCAGCCGAAAGGCCCUUGGCAGUUUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
67	Stem Replacement	Hairpin	100	100	0		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGU----------------AUACCAGCCGAAAGGCCCUUGGCAG-----------------------UUUUUU
68	Stem Replacement	Hairpin	53	56	3	9	3		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUG---------------AUACCAGCCGAAAGGCCCUUGGCAG-----------------------UUUUUU
69	Stem Replacement	Hairpin	100	100	0		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCAUACCAGCCGAAAGGCCCUUGGCAG-----------------------UUUUUU
70	Stem Replacement	Hairpin	60	70	10	5	3		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA----------------------------AUACCAGCCGAAAGGCCCUUGGCAG-------GGCACCGAGUCGGUGCUUUUUU
71	Strand displacement	Hairpin	63	68	5	0	3		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA------------------------AUCAAUACCAGCCGAAAGGCCCUUGGCAGUGAU---GGCACCGAGUCGGUGCUUUUUU
72	Strand displacement	Hairpin	81	84	4		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA-----------------------UAUCAAUACCAGCCGAAAGGCCCUUGGCAGUGAUA--GGCACCGAGUCGGUGCUUUUUU
73	Strand displacement	Hairpin	85	89	3		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA-----------------------UGUCAAUACCAGCCGAAAGGCCCUUGGCAGUGAUA--GGCACCGAGUCGGUGCUUUUUU
74	Strand displacement	Hairpin	90	89	-1		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA----------------------UUAUCAAUACCAGCCGAAAGGCCCUUGGCAGUGAUAA-GGCACCGAGUCGGUGCUUUUUU
75	Strand displacement	Hairpin	94	95	1		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA----------------------UUGUCAAUACCAGCCGAAAGGCCCUUGGCAGUGAUAA-GGCACCGAGUCGGUGCUUUUUU
76	Strand displacement	Hairpin	45	52	6		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-CG---AUACCAGCCGAAAGGCCCUUGGCAG---CGUUGGCACCGAGUCGGUGCUUUUUU
77	Strand displacement	Hairpin	73	80	7		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-AG---AUACCAGCCGAAAGGCCCUUGGCAG---CUUUGGCACCGAGUCGGUGCUUUUUU
78	Strand displacement	Hairpin	21	34	13		1		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-UG---AUACCAGCCGAAAGGCCCUUGGCAG---CAUUGGCACCGAGUCGGUGCUUUUUU
79	Strand displacement	Hairpin	90	92	2		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AAACG---AUACCAGCCGAAAGGCCCUUGGCAG--CGUUUGGCACCGAGUCGGUGCUUUUUU
80	Strand displacement	Hairpin	3	9	6	4	4		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-CGG--AUACCAGCCGAAAGGCCCUUGGCAG--CCGUUGGCACCGAGUCGGUGCUUUUUU
81	Strand displacement	Hairpin	64	80	16		2	✓	GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-AGG--AUACCAGCCGAAAGGCCCUUGGCAG--CCUUUGGCACCGAGUCGGUGCUUUUUU
82	Strand displacement	Hairpin	13	30	17		1	✓	GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-UGG--AUACCAGCCGAAAGGCCCUUGGCAG--CCAUUGGCACCGAGUCGGUGCUUUUUU
83	Strand displacement	Hairpin	87	90	4		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AAGCGG--AUACCAGCCGAAAGGCCCUUGGCAG-CCGCUUGGCACCGAGUCGGUGCUUUUUU
84	Strand displacement	Hairpin	82	79	-3		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-CGGA-AUACCAGCCGAAAGGCCCUUGGCAG-UCCGUUGGCACCGAGUCGGUGCUUUUUU
85	Strand displacement	Hairpin	8	28	20	12	4	✓	GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-CGGACAUACCAGCCGAAAGGCCCUUGGCAGGUCCGUUGGCACCGAGUCGGUGCUUUUUU
86	Strand displacement	Hairpin	95	94	-1		2		GGGGCCACUAGGACAGGAUGUUUUAGAGCUAGAAAUAGCAAGUUAAAAUAAGGCUAGUCCGUUAUCA--------------------AA-UGGACAUACCAGCCGAAAGGCCCUUGGCAGGUCCAUUGGCACCGAGUCGGUGCUUUUUU

Table 3.1: Complete results of the *in vitro* screen of rational designs. #: Design number. Strategy: The mechanism by which the design was intended to work. ”Stem Replacement” means a stem in the sgRNA was replaced by the aptamer (in some cases including a linker), with the expectation that ligand binding to the aptamer would stabilize the stem to adopt a functional sgRNA conformation. ”Induced dimerization” means the sgRNA was split in half, with each half containing part of the aptamer, in the hope that the two halves would dimerize in the presence of ligand. ”Strand displacement” means that strands were designed to base pair in two ways — one maintaining the wildtype sgRNA functional conformation and the other adopting an alternative conformation — and that ligand binding to the aptamer would stabilize the functional conformation. Domain: The domain in the sgRNA (as defined in Figure 3.1b in the main text) into which the aptamer was inserted. Cleavage: The percent of DNA that was cleaved by a design in the *in vitro* assay. The apo and holo columns refer to the cleavage with and without theophylline, respectively, Δ is the difference between these values, and σ is the standard deviation of the Δ values for designs with three or more replicates. All percentages are averages of any replicates and are rounded to the nearest integer. N: The number of replicates for each design. Active: We denote a design as “active” if it exhibited a >15% change in cleavage in response to ligand. Sequence: The sequence of the design, including the AAVS spacer used in this assay (Table 3.4). Sequences are aligned and color coded by sgRNA domain as in Figure 3.1b (grey: spacer; blue: upper stem; navy: nexus; teal: hairpin; purple: aptamer; orange: other regions).

#	Domain	Size	Sequence
1	Upper Stem	4^{10}	GTTTTANNNN--ATACCAGCCGAAAGGCCCTTGGCAG--NNNNNNTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
2	Upper Stem	4^{11}	GTTTTANNNN--ATACCAGCCGAAAGGCCCTTGGCAG-NNNNNNNTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
3	Upper Stem	4^{12}	GTTTTANNNN--ATACCAGCCGAAAGGCCCTTGGCAGNNNNNNNNTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
4	Upper Stem	4^{11}	GTTTTANNNNN-ATACCAGCCGAAAGGCCCTTGGCAG--NNNNNNTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
5	Upper Stem	4^{12}	GTTTTANNNNN-ATACCAGCCGAAAGGCCCTTGGCAG-NNNNNNNTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
6	Upper Stem	4^{12}	GTTTTANNNNNNATACCAGCCGAAAGGCCCTTGGCAG--NNNNNNTAAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
7	Nexus	4^{4}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANN---ATACCAGCCGAAAGGCCCTTGGCAG---NNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
8	Nexus	4^{5}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANN---ATACCAGCCGAAAGGCCCTTGGCAG--NNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
9	Nexus	4^{6}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANN---ATACCAGCCGAAAGGCCCTTGGCAG-NNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
10	Nexus	4^{7}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANN---ATACCAGCCGAAAGGCCCTTGGCAGNNNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
11	Nexus	4^{5}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNN--ATACCAGCCGAAAGGCCCTTGGCAG---NNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
12	Nexus	4^{6}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNN--ATACCAGCCGAAAGGCCCTTGGCAG--NNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
13	Nexus	4^{7}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNN--ATACCAGCCGAAAGGCCCTTGGCAG-NNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
14	Nexus	4^{8}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNN--ATACCAGCCGAAAGGCCCTTGGCAGNNNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
15	Nexus	4^{6}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNN-ATACCAGCCGAAAGGCCCTTGGCAG---NNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
16	Nexus	4^{7}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNN-ATACCAGCCGAAAGGCCCTTGGCAG--NNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
17	Nexus	4^{8}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNN-ATACCAGCCGAAAGGCCCTTGGCAG-NNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
18	Nexus	4^{9}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNN-ATACCAGCCGAAAGGCCCTTGGCAGNNNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
19	Nexus	4^{7}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNNNATACCAGCCGAAAGGCCCTTGGCAG---NNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
20	Nexus	4^{8}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNNNATACCAGCCGAAAGGCCCTTGGCAG--NNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
21	Nexus	4^{9}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNNNATACCAGCCGAAAGGCCCTTGGCAG-NNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
22	Nexus	4^{10}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAANNNNNATACCAGCCGAAAGGCCCTTGGCAGNNNNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
23	Hairpin	4^{10}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAAGGCTAGTCCGTTNN--NNNNATACCAGCCGAAAGGCCCTTGGCAG--NNNNGGCACCGAGTCGGTGCTTTTTT
24	Hairpin	4^{11}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAAGGCTAGTCCGTTNN--NNNNATACCAGCCGAAAGGCCCTTGGCAG-NNNNNGGCACCGAGTCGGTGCTTTTTT
25	Hairpin	4^{12}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAAGGCTAGTCCGTTNN--NNNNATACCAGCCGAAAGGCCCTTGGCAGNNNNNNGGCACCGAGTCGGTGCTTTTTT
26	Hairpin	4^{11}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAAGGCTAGTCCGTTNNN-NNNNATACCAGCCGAAAGGCCCTTGGCAG--NNNNGGCACCGAGTCGGTGCTTTTTT
27	Hairpin	4^{12}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAAGGCTAGTCCGTTNNN-NNNNATACCAGCCGAAAGGCCCTTGGCAG-NNNNNGGCACCGAGTCGGTGCTTTTTT
28	Hairpin	4^{12}	GTTTTAGAGCTA-----GAAA-----TAGCAAGTTAAAATAAGGCTAGTCCGTTNNNNNNNNATACCAGCCGAAAGGCCCTTGGCAG--NNNNGGCACCGAGTCGGTGCTTTTTT
29	Hairpin	4^{9}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNNNCCNN-ATNNGCCGATACCAGCCGAAAGGCCCTTGGCAG--CGACGGCACCGAGTCGGTGCTTTTTT
30	Hairpin	4^{10}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNNNCCNNNATNNGCCGATACCAGCCGAAAGGCCCTTGGCAG--CGACGGCACCGAGTCGGTGCTTTTTT
31	Hairpin	4^{10}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNN--TCCNNNNNN-GCCGTCGGGGTGCCCTTCTGCGTGAAGGCTGAGAAATACCCGTATCACCTGATCTGGATAATGCCAGCGTAGGGAACGACGGCACCGAGTCGGTGCTTTTTT
32	Hairpin	4^{11}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNN--TCCNNNNNNNGCCGTCGGGGTGCCCTTCTGCGTGAAGGCTGAGAAATACCCGTATCACCTGATCTGGATAATGCCAGCGTAGGGAACGACGGCACCGAGTCGGTGCTTTTTT
33	Hairpin	4^{11}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNNN-TCCNNNNNN-GCCGTCGGGGTGCCCTTCTGCGTGAAGGCTGAGAAATACCCGTATCACCTGATCTGGATAATGCCAGCGTAGGGAACGACGGCACCGAGTCGGTGCTTTTTT
34	Hairpin	4^{12}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNNN-TCCNNNNNNNGCCGTCGGGGTGCCCTTCTGCGTGAAGGCTGAGAAATACCCGTATCACCTGATCTGGATAATGCCAGCGTAGGGAACGACGGCACCGAGTCGGTGCTTTTTT
35	Hairpin	4^{12}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGNNNNNNTCCNNNNNN-GCCGTCGGGGTGCCCTTCTGCGTGAAGGCTGAGAAATACCCGTATCACCTGATCTGGATAATGCCAGCGTAGGGAACGACGGCACCGAGTCGGTGCTTTTTT
36	Upper Stem	4^{8}	GTTTCANNNGATACCAGCCGAAAGGCCCTTGGCAGCNNNNN--TGAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
37	Upper Stem	4^{9}	GTTTCANNNGATACCAGCCGAAAGGCCCTTGGCAGCNNNNNN-TGAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
38	Upper Stem	4^{10}	GTTTCANNNGATACCAGCCGAAAGGCCCTTGGCAGCNNNNNNNTGAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
39	Nexus	4^{5}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAANNN--ATACCAGCCGAAAGGCCCTTGGCAG--TNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
40	Nexus	4^{7}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAANNNN-ATACCAGCCGAAAGGCCCTTGGCAG-NTNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
41	Nexus	4^{9}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAANNNNNATACCAGCCGAAAGGCCCTTGGCAGNNTNNGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
42	Hairpin	4^{8}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGCTAGTCCGTTATCANNNN-ATACCAGCCGAAAGGCCCTTGGCAGNNNN-GGCACCGAGTCGGTGCTTTTTT
43	Hairpin	4^{9}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGCTAGTCCGTTATCANNNN-ATACCAGCCGAAAGGCCCTTGGCAGNNNNNGGCACCGAGTCGGTGCTTTTTT
44	Hairpin	4^{9}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGCTAGTCCGTTATCANNNNNATACCAGCCGAAAGGCCCTTGGCAGNNNN-GGCACCGAGTCGGTGCTTTTTT
45	Hairpin	4^{10}	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGCTAGTCCGTTATCANNNNNATACCAGCCGAAAGGCCCTTGGCAGNNNNNGGCACCGAGTCGGTGCTTTTTT

Table 3.2: Library sequences screened by FACS (see Methods). ligRNA^{+} was selected from library #29, while ligRNA^{-} was selected from library #22. Libraries #29 and #30 are based on the only clone isolated from library #26. Color coding of sgRNA domains is as in Table 3.1. No functional ligRNAs were isolated from libraries #1-6 because the only initial hits isolated from these libraries were spacer dependent.

Name	Library	Picks	Fold Change				Sequence
			sgG1	sgG2	sgR1	sgR2	
	1–6	1/18	10.2^{+}	—	1.0^{-}	—	GUUUUA--AAGGAUACCAGCCGAAAGGCCCUUGGCAGCUUUAGC-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	2/18	12.4^{+}	—	1.2^{+}	—	GUUUUA-CCCGAAUACCAGCCGAAAGGCCCUUGGCAGUCUUCGC-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	10.7^{+}	—	1.4^{+}	—	GUUUUA--AUCGAUACCAGCCGAAAGGCCCUUGGCAGCGGCUU--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	11.3^{+}	—	1.1^{-}	—	GUUUUA--UAGGAUACCAGCCGAAAGGCCCUUGGCAGCUGCUCGCUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	9.0^{+}	—	1.1^{+}	—	GUUUUA-ACCUGAUACCAGCCGAAAGGCCCUUGGCAGCUGUAGA-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	8.0^{+}	—	1.0^{+}	—	GUUUUA-CCGAGAUACCAGCCGAAAGGCCCUUGGCAGCCAGUCU-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	7.5^{+}	—	1.4^{-}	—	GUUUUA--CCCGAUACCAGCCGAAAGGCCCUUGGCAGCGAGAGCUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	8.2^{+}	—	1.3^{-}	—	GUUUUA-CGCGGAUACCAGCCGAAAGGCCCUUGGCAGCCUAGG--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	7.7^{+}	—	1.0^{+}	—	GUUUUA-GCUGAAUACCAGCCGAAAGGCCCUUGGCAGUCGGGCU-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	9.4^{+}	—	1.1^{-}	—	GUUUUA--CGUGAUACCAGCCGAAAGGCCCUUGGCAGCUUAUGC-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	8.6^{+}	—	1.0^{+}	—	GUUUUA--UCGCAUACCAGCCGAAAGGCCCUUGGCAGGCCUGC--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	6.0^{+}	—	1.2^{-}	—	GUUUUA--CCAGAUACCAGCCGAAAGGCCCUUGGCAGCUGCCAGCUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	6.4^{+}	—	1.3^{-}	—	GUUUUA--AUCGAUACCAGCCGAAAGGCCCUUGGCAGCGCUGCUUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	33.6^{+}	—	1.3^{+}	—	GUUUUA--GCGGAUACCAGCCGAAAGGCCCUUGGCAGCCGGGCUUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	17.9^{+}	—	1.2^{-}	—	GUUUUA--ACGGAUACCAGCCGAAAGGCCCUUGGCAGCCGGGCAUUAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/18	8.0^{+}	—	1.2^{-}	—	GUUUUA--AUAGAUACCAGCCGAAAGGCCCUUGGCAGCAGCAGC-UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/21	6.6^{-}	—	—	—	GUUUUAAAGCUGAUACCAGCCGAAAGGCCCUUGGCAGCGGCGC--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/21	6.0^{-}	—	—	—	GUUUUA-CCCUAAUACCAGCCGAAAGGCCCUUGGCAGUGGGGU--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	10/21	6.9^{-}	—	1.0^{+}	—	GUUUUAGCGCUGAUACCAGCCGAAAGGCCCUUGGCAGCGUCGC--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	1/21	3.9^{-}	—	—	—	GUUUUAGUCUGUAUACCAGCCGAAAGGCCCUUGGCAGAUAGAU--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	3/21	3.7^{-}	—	—	—	GUUUUAACCUGCAUACCAGCCGAAAGGCCCUUGGCAGGUAGGU--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	2/21	7.1^{-}	—	—	—	GUUUUA-ACCUCAUACCAGCCGAAAGGCCCUUGGCAGGUGGU---UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	1–6	3/21	5.6^{-}	—	—	—	GUUUUA-CCCCCAUACCAGCCGAAAGGCCCUUGGCAGGGGGGU--UAAAAUAAGGCUAGUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	4/20	8.6^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGGGGAAUACCAGCCGAAAGGCCCUUGGCAGUCUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
*	7–22	5/20	13.7^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGUGGGAUACCAGCCGAAAGGCCCUUGGCAGCCUACGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
ligRNA^{-}	—	—	15.8^{-}	4.3^{-}	5.5^{-}	10.2^{-}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGUGGGAUACCAGCCGAAAGGCCCUUGGCAGCCUACGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	4/20	10.2^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGAAGGAUACCAGCCGAAAGGCCCUUGGCAGCUUUCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	1/20	13.2^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGGUG-AUACCAGCCGAAAGGCCCUUGGCAGCAUCUGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	1/20	11.0^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGGGGAAUACCAGCCGAAAGGCCCUUGGCAGUCUUCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	1/20	—	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGGGGAAUACCAGCCGAAAGGCCCUUGGCAGUUUGCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	2/20	5.6^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGCCCGAUACCAGCCGAAAGGCCCUUGGCAGCGAGUGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	1/20	7.3^{-}	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGGGAAAUACCAGCCGAAAGGCCCUUGGCAG-UUCCGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	7–22	1/20	—	—	—	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGUG--AUACCAGCCGAAAGGCCCUUGGCAG--CACGUUAUCAACUUGAAAAAGUGGCACCGAGUCGGUGCUUUUUU
	23–28	6/6	14.1^{+}	—	1.4^{+}	—	GUUUUAGAGCUA-----GAAA-----UAGCAAGUUAAAAUAAGGCUAGUCCCUU-UUCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	9.7^{+}	—	4.8^{+}	4.4^{+}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGCGGUCCCGUCAUCAGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	4.5^{+}	1.3^{+}	5.4^{+}	—	GUUUCAGAGCUAUGCUUGAAACAGCAUAGCAAGUUGAAAUAAGGACGAACCGUAAUCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	9.6^{+}	1.7^{+}	3.8^{+}	—	GUUUCAGAGCUUUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGUCUGACCAGA-UCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	7.3^{+}	1.3^{+}	9.8^{+}	4.5^{+}	GUUUCAGAGCUUUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGUGACACCGCA-UCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	10.1^{+}	1.3^{+}	2.5^{+}	6.5^{+}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGUGGUACCAUA-UCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	7.5^{+}	1.4^{-}	7.6^{+}	6.1^{+}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGUAAACCCCUCAUCAGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	2/15	10.8^{+}	—	3.4^{+}	—	GUUUCAGAGCUAUGCUGGAAACAGAAUAGCAAGUUGAAAUAAGGAUCCUCCCGCAUGCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	10.8^{+}	1.1^{-}	5.0^{+}	6.8^{+}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGACGUCCCGCA-UUCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	7.7^{+}	—	4.8^{+}	—	GUUUCAGAGCUAUGCUGGAAACAGUAUAGCAAGUUGAAAUAAGGUUCGUCCGCCAUUCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
ligRNA^{+}	29–30	1/15	10.3^{+}	1.4^{-}	8.6^{+}	7.8^{+}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGGUGUCCCGUA-UACGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	7.2^{+}	—	3.8^{+}	3.8^{+}	GUUUCAGAGCUAUACUGGAAACAGCAUAGCAAGUUGAAAUAAGGGCACUCCUAA-UCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
ligRNA$^{+}_{2}$	29–30	1/15	7.2^{+}	1.0^{-}	15.3^{+}	10.8^{+}	GUUUCAGAGC-AUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGUCUUCCCCGCAUCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	7.2^{+}	1.4^{-}	10.3^{+}	—	GUUUCAGAGCCAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGACGGUCCCGCAUCCGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
	29–30	1/15	11.2^{+}	1.2^{-}	3.2^{+}	5.3^{+}	GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGGACUCUCCGUA-UCGGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU

Table 3.3: Ligand-sensitive sgRNAs isolated from FACS screen. Name: Names of designs described in the main text. The asterisk (*) denotes the design that was the precursor to ligRNA$^-$; we created ligRNA$^-$ by transferring the nexus stem from this precursor into the optimized sgRNA scaffold described in ref. [1]. Library: Libraries (Table 3.2) from which each design was isolated. Picks: The number of times each design appeared (first number) out of the total number of colonies picked to sequence from each library (second number). Fold Change: The fold change in fluorescence in response to the addition of theophylline, measured by flow cytometry. All data in this table are single-replicate. The subcolumns (sgG1, sgG2, sgR1, sgR2) indicate the spacer used for each measurement. Superscripts indicate whether CRISPRi was activated (+) or inhibited (−) by theophylline. Dashes indicate spacers that were not tested. Note that the designs selected from libraries 1–6 appear to be spacer-dependent, i.e. they are functional with the sgG1 spacer that was used in the selection but not with the sgR1 spacer (which for these libraries was not used in the selection). We therefore did not pursue these designs further. In contrast, the designs from libraries 29–30 were selected using both the sgG1 and sgR1 spacers as described in Figure 3.2c, and in validation experiments also function with the sgR2 spacer (the sgG2 spacer is not functional with any of the designs). Sequence: The sequence of the isolated design, color coded as in Table 3.1. Note that the randomized regions on either side of the aptamer often form based-paired stems. Also note that for the nexus insertions (libraries 7–22), U95 and G63 (in ligRNA$^-$ numbering, alternatively the third position from the 3' end of the nexus and the position at the 5' end of the nexus) are conserved in 17 and 20 of the 20 of the isolated sequences, respectively. This observation corroborates the importance of these positions in our mutagenesis experiments (Figure 3.6).

Name	Sequence
T7 promoter	TATAGTAATAATACGACTCACTATAG
AAVS spacer	GGGGCCACTAGGGACAGGAT
sgG1 spacer	CATCTAATTCAACAAGAATT
sgR1 spacer	AACTTTCAGTTTAGCGGTCT
sgG2 spacer	AGTAGTGCAAATAAATTTAA
sgR2 spacer	TGGAACCGTACTGGAACTGC
Theophylline aptamer	ATACCAGCCGAAAGGCCCTTGGCAG
3-Methylxanthine aptamer	ATACCAGCCGAAAGGCCATTGGCAG
Positive control	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
Negative control	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAACCCTAGTCCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
ligRNA$^+$	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGG-TGTCCCGTATACGCCGATACCAGCCGAAAGGCCCTTGGCAGCGACGGCACCGAGTCGGTGCTTTTTT
ligRNA$^+_2$	GTTTCAGAGC-ATGCTGGAAACAGCATAGCAAGTTGAAATAAGGTCTTCCCCGCATCCGCCGATACCAGCCGAAAGGCCCTTGGCAGCGACGGCACCGAGTCGGTGCTTTTTT
ligRNA$^+_3$	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGGC-TGTGCCGTATACGCCGATACCAGCCGAAAGGCCCTTGGCAGCGACGGCACCGAGTCGGTGCTTTTTT
ligRNA$^+_4$	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGCG-TGTCGCGTATACGCGGATACCAGCCGAAAGGCCCTTGGCAGCCACGGCACCGAGTCGGTGCTTTTTT
ligRNA$^-$	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGTGGGATACCAGCCGAAAGGCCCTTGGCAGCCTACGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
ligRNA$^-_2$	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGAGGGATACCAGCCGAAAGGCCCTTGGCAGCCTTCGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT
ligRNA$^-_3$	GTTTCAGAGCTATGCTGGAAACAGCATAGCAAGTTGAAATAAGTGGAATACCAGCCGAAAGGCCCTTGGCAGTCTACGTTATCAACTTGAAAAAGTGGCACCGAGTCGGTGCTTTTTT

Table 3.4: DNA sequences of main constructs.

			pos (% cut)				neg (% cut)				ligRNA$^+$ (% cut)				ligRNA$^-$ (% cut)				
#	Spacer (with context)	Score	*apo*	*holo*	Δ	σ	*apo*	*holo*	Δ	σ	*apo*	*holo*	Δ	σ	*apo*	*holo*	Δ	σ	N
1	gcagGGGAACTCAAGAGCGGAGGGtggtca	0.622	91	90	-1	1	1	2	1	0	20	57	37	6	89	15	-74	3	3
2	acgcGGGTCTTACCTTTAATAGAAggggtc	0.523	94	94	0	2	3	2	-1	1	36	88	52	10	94	30	-64	13	3
3	cccgGGGTTATGCCTACGATGACCgggaat	0.582	95	95	-0	1	3	1	-2	2	43	79	36	17	93	24	-69	17	3
4	atcaGGGTGTAAGGATCGGAGTGCaggctc	0.518	91	92	1	3	0	0	0	0	7	30	23	6	87	8	-79	9	3
5	ctgtGGGATAAGGAGTTCTCGTGTaggggc	0.568	—	—	–	–	0	1	0	0	47	81	34	22	95	30	-65	25	3
6	gcagGGGCTTGTATTGGTTCTGAGtggata	0.605	90	89	-1	1	2	1	-1	1	27	81	54	5	90	44	-47	22	3
7	gaagGGGAAAGAACTCATTATCGTtggtgg	0.587	97	97	-0	0	3	2	-2	1	52	77	26	9	97	41	-56	4	3
8	ctttGGGCGGGTTGAATAGTCGTTgggtga	0.505	91	92	1	4	0	0	-0	0	28	71	42	11	69	24	-45	10	3
9	ttgaGGGCATGACCTAAACTCTATcggcaa	0.582	94	93	-1	1	4	2	-2	2	2	16	14	3	90	15	-75	4	3
10	taccGGGTTGTGATTGGTAACAGAtggggg	0.563	96	95	-0	1	0	0	-0	1	44	66	21	3	87	26	-61	7	3
11	acaaGGGACCATATAGTAGAACATgggcca	0.702	93	93	-0	1	5	3	-2	2	28	59	31	10	93	47	-46	19	4
12	tctcGGGGTCATCAGTATGAGTAAagggac	0.548	83	85	2	3	3	1	-2	2	35	58	22	6	89	33	-56	9	4
13	cgctGGGGATAACAAATGACCATGcggtct	0.728	97	97	-0	0	1	1	-1	1	46	64	19	17	71	22	-49	14	3
14	acgaGGGAGCTCACTATTCCAGGTcggtaa	0.688	94	94	-0	0	0	0	0	0	8	32	25	14	34	3	-31	14	3
15	gagaGGGTCAGACGGGAAGCCGGAcggtat	0.627	89	89	0	2	0	0	0	0	2	15	14	3	71	25	-46	16	3
16	gtacGGGTTAACGGATCCAACACGaggacg	0.708	97	97	-0	0	0	0	0	0	4	15	11	1	45	2	-43	7	3
17	ttagGGGTGCCTTTCCCACGAGCTgggtag	0.515	96	95	-2	2	0	0	0	0	5	24	19	1	1	3	2	0	3
18	ggccGGGCAGGGGCGTGAGAATCGtgggaa	0.522	90	90	1	1	0	0	-0	0	7	23	16	10	39	19	-20	16	3
19	acatGGGGGTATTCCGACTTGACAtggtgg	0.594	91	90	-0	0	0	0	-0	0	26	37	12	6	46	20	-26	7	3
20	gcagGGGAGTGGGACAGGAGTACGtgggaa	0.525	92	90	-2	0	0	0	0	0	3	9	6	3	46	9	-37	14	3
21	tatcGGGTTGGACTCTCTAACGATaggaaa	0.622	97	96	-1	1	0	0	0	0	12	18	6	4	50	8	-42	13	3
22	caccGGGCCAGCGTCGTGCAAGCGggggtg	0.510	94	91	-3	4	0	0	0	0	0	3	3	4	22	1	-21	12	3
23	gtgtGGGGTTAGAGGCTGAGTCCAgggtcc	0.534	95	94	-1	1	0	0	0	0	2	8	6	1	0	0	0	0	3
24	tcgaGGGCCCGGAGCCTGGCACGTtggaag	0.527	96	95	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	2

Table 3.5: Detailed results for the *in vitro* DNA cleavage assay in the context of different spacers (Figure 3.2g in the main text). #: The spacer number in Figure 3.2g. Spacer (with context): In upper case is the 20 nt spacer sequence being tested. Note that each spacer begins with 3 Gs for facile transcription by T7 polymerase. In upper and lower case is the corresponding sequence present in the target DNA. This includes 10 bp of context and (to the right of the spacer) the NGG protospacer adjacent motif (PAM). These sequences were chosen as described in the Methods section. Score: The "Rule Set 2" score of the target sequence by the method of ref. [2], which predicts Cas9 cleavage efficiency for the target sequence. We only tested sequences with scores greater than 0.5 (approximately half of those generated). pos, neg, ligRNA$^+$, ligRNA$^-$: Cleavage data for the controls and indicated sgRNA. The apo, holo, Δ, σ, N columns are as in Table 3.1.

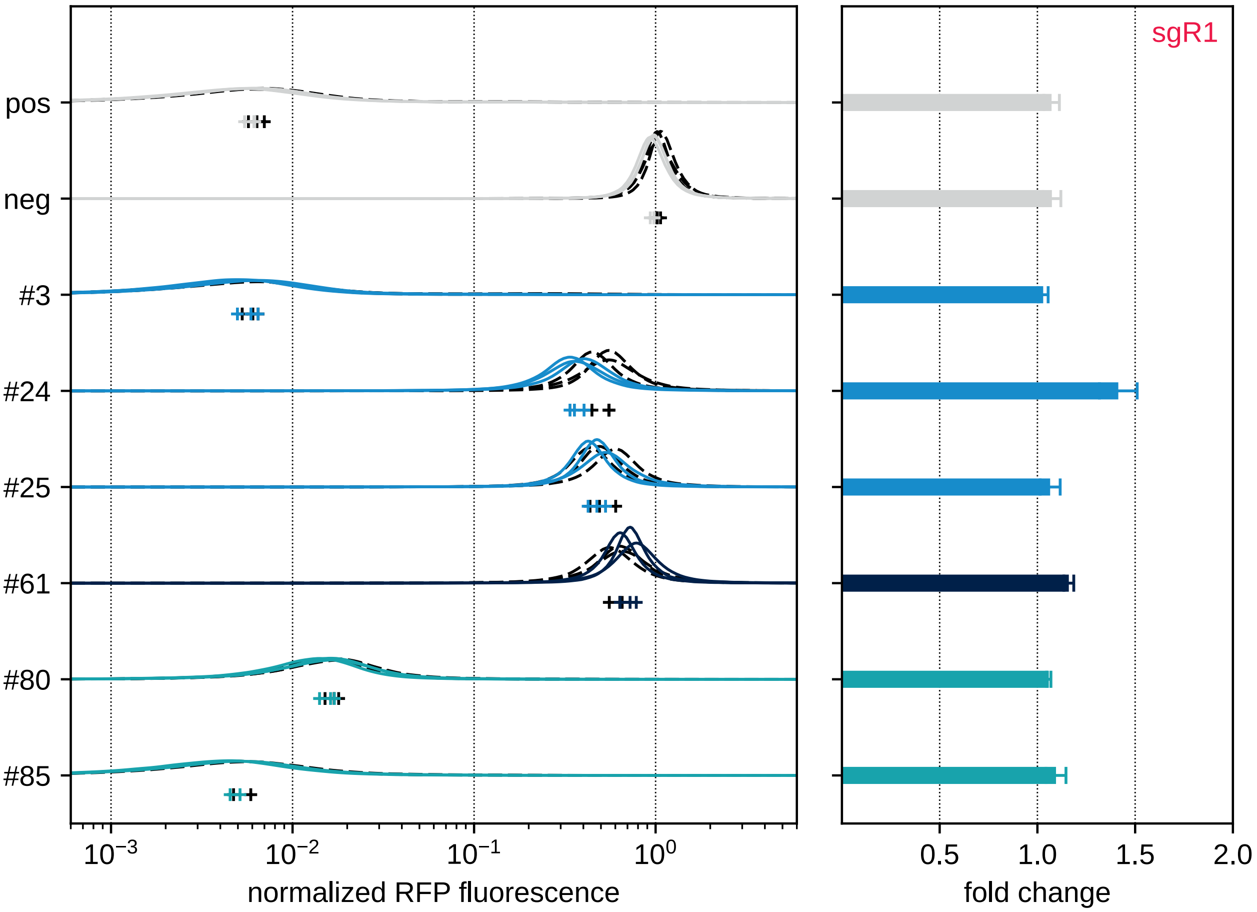

Figure 3.4: The strongest rational designs (determined by the *in vitro* cleavage assay, Table 3.1) have weak ligand-sensitivity in a CRISPRi-based assay in E. coli. Flow cytometry traces and fold changes for the rational designs that were tested in E. coli. The labels on the y-axis refer to Table 3.1. Traces are color-coded by aptamer insertion site (defined in Figure 3.1b). RFP fluorescence values for each cell are normalized by both GFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All other lines and symbols are as described in Figure 3.2. Data are from three experiments performed on the same day (for all other experiments reported in this paper, replicates were performed on different days). Designs #24 and #61 display a small ligand-dependency. For design #24 adding theophylline shifts the fluorescence distribution to the left (less fluorescence, indicating stabilization of the functional sgRNA conformation with ligand), whereas for design #61 the opposite is the case (shift to more fluorescence indicating destabilization of the functional sgRNA conformation with ligand). These changes are in the same direction as observed for these designs in the *in vitro* DNA cleavage assay (Table 3.1).

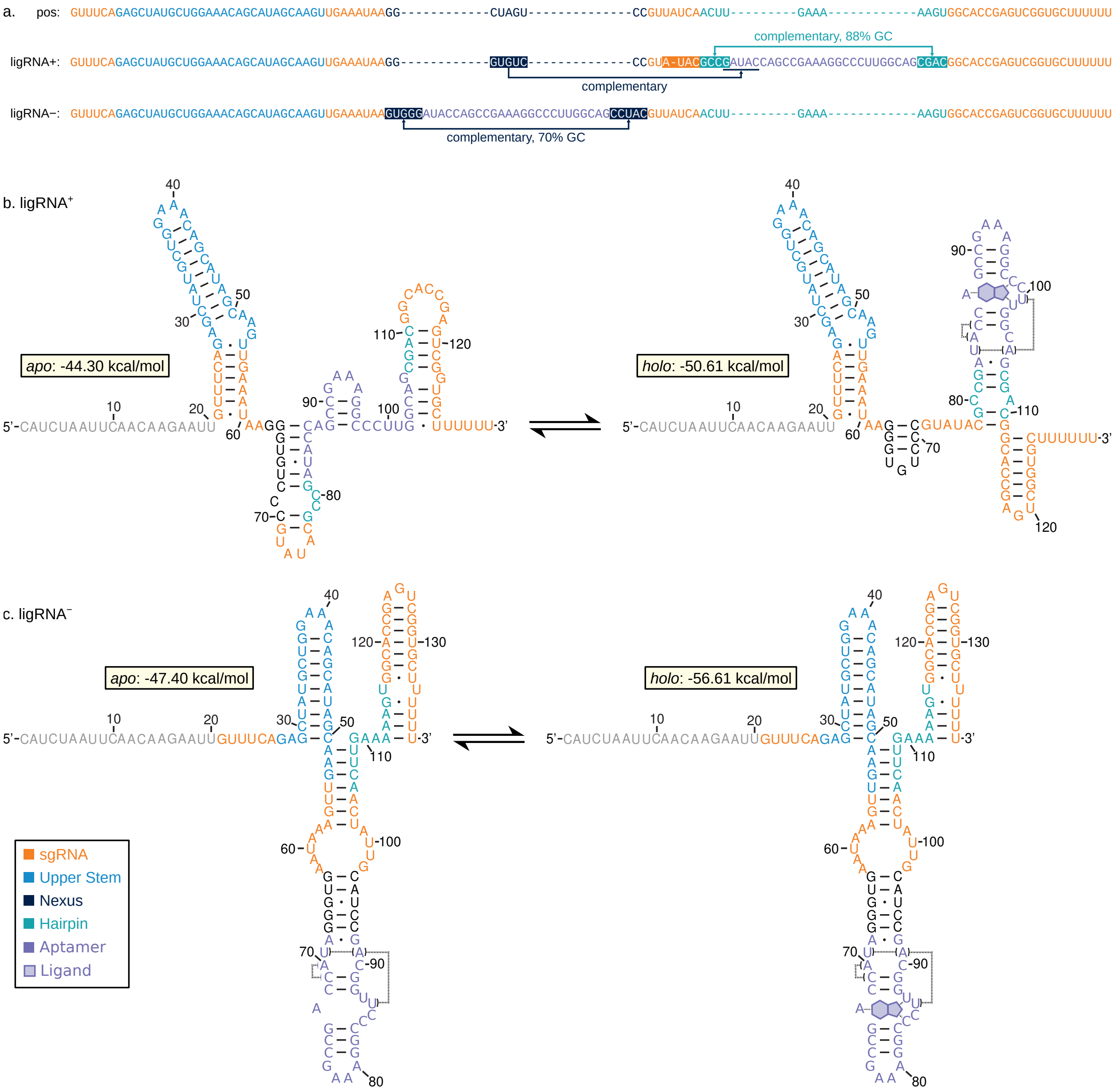
a.
pos: GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGG------------CUAGU-------------CCGUUAUCAACUU---------GAAA------------AAGUGGCACCGAGUCGGUGCUUUUUU
complementary, 88% GC
ligRNA+: GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGG------------GUGUC-------------CCGUA-UACGCCGAUACCAGCCGAAAGGCCCUUGGCAGCGACGGCACCGAGUCGGUGCUUUUUU
complementary
ligRNA−: GUUUCAGAGCUAUGCUGGAAACAGCAUAGCAAGUUGAAAUAAGUGGGAUACCAGCCGAAAGGCCCUUGGCAGCCUACGUUAUCAACUU---------GAAA------------AAGUGGCACCGAGUCGGUGCUUUUUU
complementary, 70% GC
b. ligRNA+
apo: -44.30 kcal/mol
holo: -50.61 kcal/mol
5'-CAUCUAAUUCAACAAGAAUU
-3'
c. ligRNA−
apo: -47.40 kcal/mol
holo: -56.61 kcal/mol
sgRNA
Upper Stem
Nexus
Hairpin
Aptamer
Ligand

Figure 3.5: Secondary structure predictions suggest mechanisms of ligand sensitivity. (a) Sequence alignment of the positive control sgRNA, ligRNA$^+$, and ligRNA$^-$. Nucleotides are color-coded by domain, and randomized positions are shaded. Several sequence features of the selected ligRNAs are noted. (b,c) Secondary structure and free energy predictions for ligRNA$^+$ (b) and ligRNA$^-$ (b) in both the apo and holo states, calculated as described in the Methods section. Nucleotides are color-coded by domain. (b) In the apo state of ligRNA$^+$, the nexus is predicted to base-pair with the aptamer, but in the holo state, the sgRNA is predicted to fold correctly. (c) The apo and holo states for ligRNA$^-$ are predicted to have the same fold and neither prediction recapitulates the known sgRNA stems. This figure was drawn by Kyle Watters.

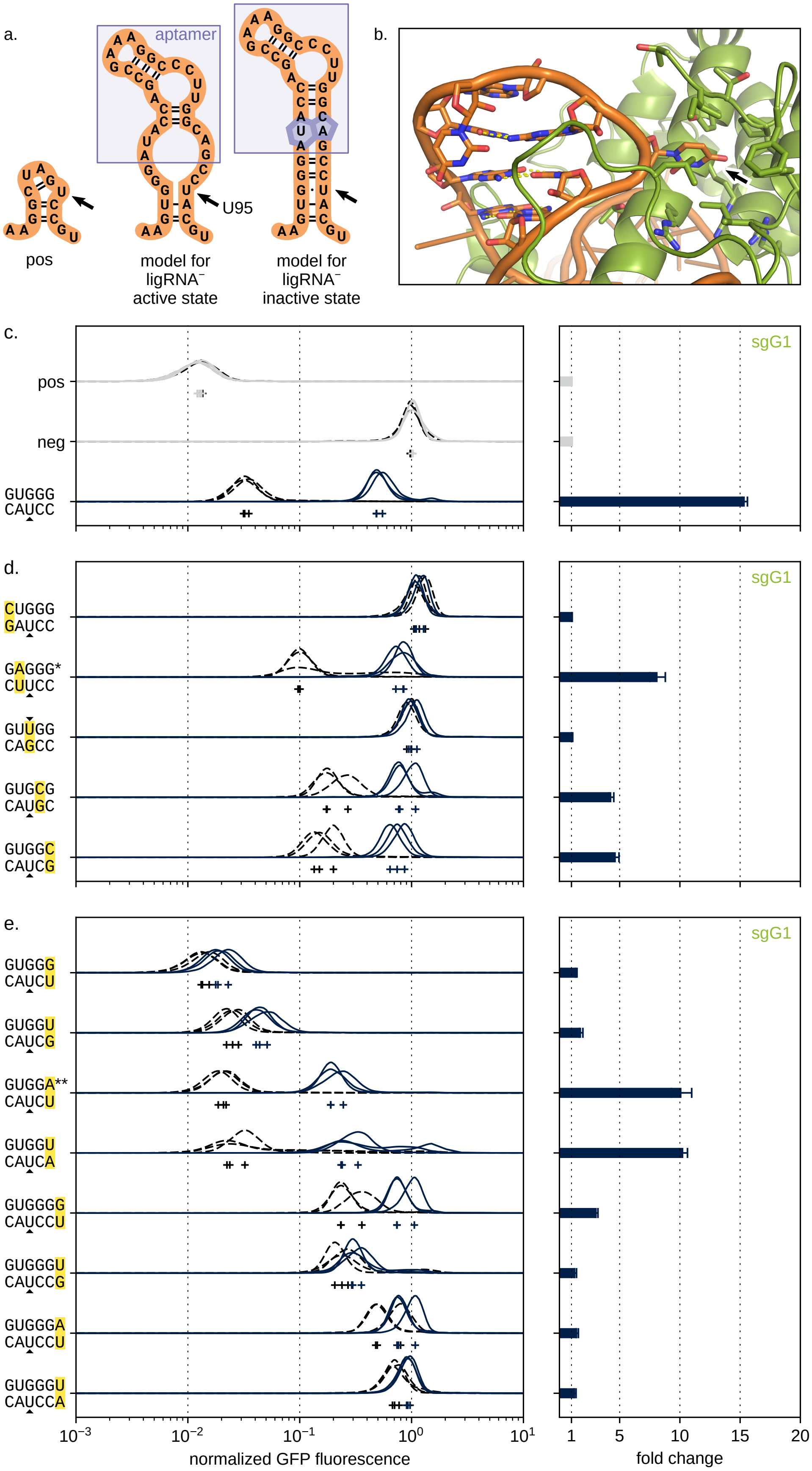
a.
aptamer
U95
pos
model for ligRNA⁻ active state
model for ligRNA⁻ inactive state
b.
c.
pos
neg
GUGGG
CAUCC
sgG1
d.
CUGGG
GAUCC
GAGGG*
CUUCC
GUUGG
CAGCC
GUGCG
CAUGC
GUGGC
CAUCG
e.
GUGGG
CAUCU
GUGGU
CAUCG
GUGGA**
CAUCU
GUGGU
CAUCA
GUGGGG
CAUCCU
GUGGGU
CAUCCG
GUGGGA
CAUCCU
GUGGGU
CAUCCA
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
normalized GFP fluorescence
1
5
10
15
20
fold change

Figure 3.6: Mechanistic insights into ligRNA$^-$ function. (a) Model of possible mechanism, where ligRNA$^-$ functions by sequestering the indicated uracil (U95) in the presence of the ligand. U95 is unpaired in wildtype sgRNA (left). Our hypothesis is that in ligRNA$^-$ U95 is unpaired to a larger extent in the apo state (center) than in the holo state (right). (b) A crystal structure of Cas9 in complex with an sgRNA [3] (PDB ID 4UN3) shows the indicated uracil flipped out (black arrow) and interacting with the Cas9 protein. (c-e) Flow cytometry traces and fold changes for different mutants of ligRNA$^-$. The labels show the sequence of the particular mutant being tested. Mutations relative to ligRNA$^-$ are highlighted in yellow. The uracil in question is indicated with a small triangle. The ligRNA$^-_2$ and ligRNA$^-_3$ variants (Table 3.4; Figure 3.10) are marked by * and **, respectively. GFP fluorescence values for each cell are normalized by both RFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All other lines and symbols are described in Figure 3.2. (c) Positive and negative controls, and ligRNA$^-$. (d) Strand-swap mutations for each position along the nexus stem. None of the mutants are as functional as ligRNA$^-$, but the design is tolerant to strand-swap mutation at positions 2, 4, and 5. As expected, position 3 containing the critical uracil is intolerant to mutation. The simple model in (a) does not explain the intolerance of position 1 however. (e) Modulating the strength of the base-pairs between the uracil and the aptamer has a predictable effect on function. From top to bottom, the mutants are arranged in the order of increasing base-pairing strength. Fluorescence distributions shifted to the left indicate stronger activation of the sgRNA (fluorescence is more effectively repressed). Our hypothesis (that ligRNA$^-$ works by sequestering the uracil upon ligand binding) predicts that weakening or strengthening the base-pairs between the uracil and the aptamer should increase or decrease activation, respectively. The clear downward diagonal trend in the populations supports this hypothesis and demonstrates that we can tune the dynamic range of ligRNA$^-$ to some extent. The third and fourth mutants (AU and UA) may be useful for applications where strong repression is desired because they show stronger repression that the original ligRNA$^-$ design in the absence of the ligand, although their dynamic range (10x) is somewhat narrower than that of ligRNA$^-$ (15x).

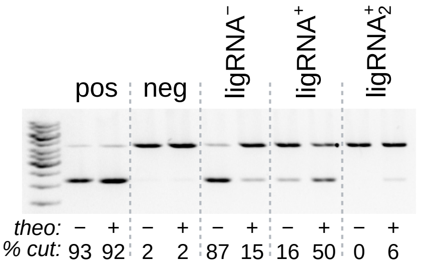

Figure 3.7: Representative gel from the *in vitro* spacer assay. Shown is a single replicate for spacer #1 (Figure 3.2g). The upper and lower bands are uncleaved and cleaved DNA, respectively. Each design was tested in the absence and presence of theophylline. The amount of DNA cleavage was quantified by gel densiometry and is reported as a percentage below each lane.

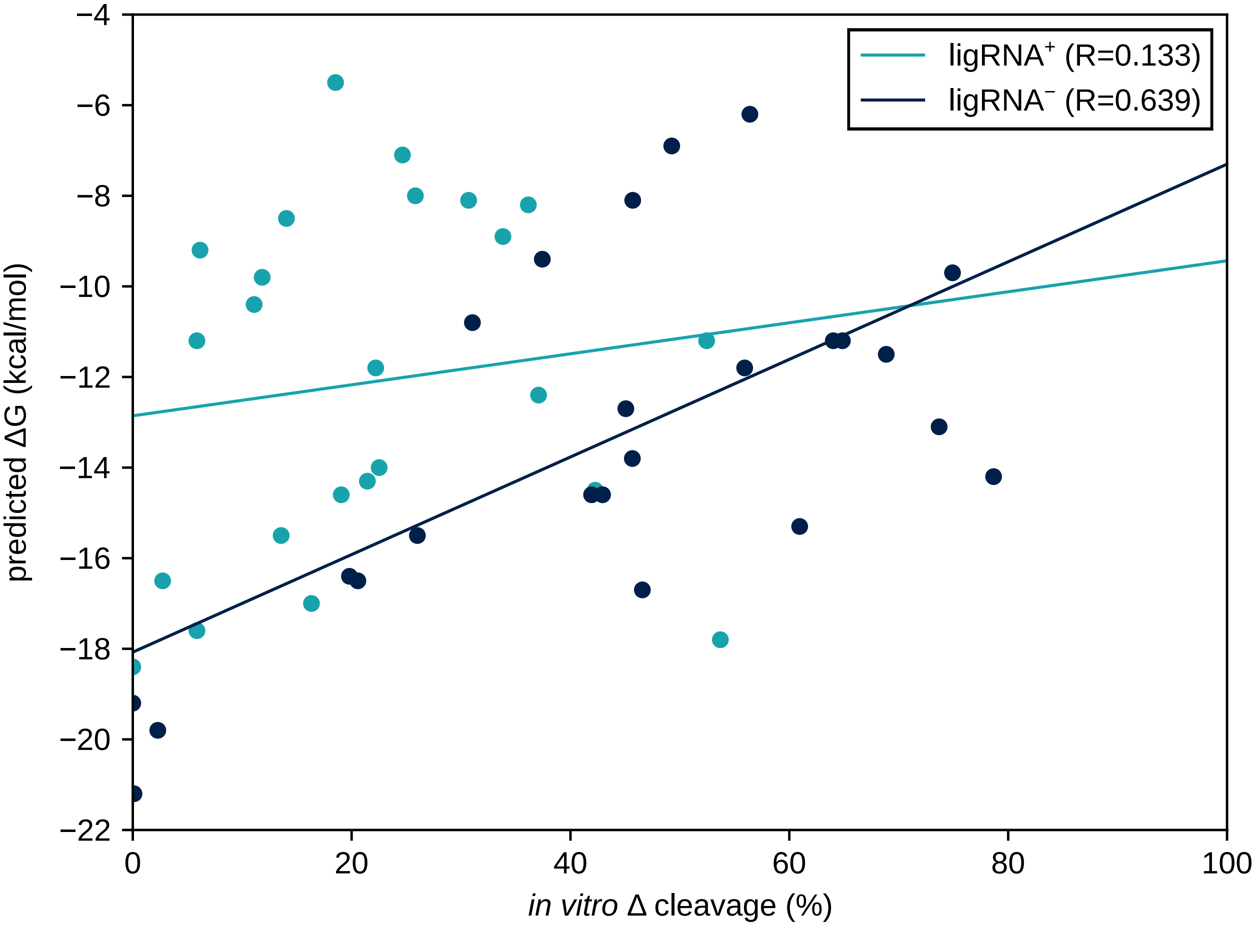

Figure 3.8: Correlation between ligRNA function and the predicted binding free energy of base-pairing between the spacer and the aptamer insert. Binding free energies (y-axis) were calculated using the duplexfold method from the python3 API of the ViennaRNA package (version 2.4.3). This method returns the minimum binding energy between two strands of RNA considering only inter-strand base pairs. For each calculation, the first strand was one of the 24 20 nt spacers used in the *in vitro* spacer assay (Table 3.5). The second strand was GCCGAUACCAGCCGAAAGGC-CCUUGGCAGCGAC for ligRNA$^+$ or GUGGGAUACCAGCCGAAAGGCCCUUGGCAGCCUAC for ligRNA$^-$. These sequences include both the aptamer and the randomized linker connecting the aptamer to the sgRNA scaffold. Percent cleavage values (x-axis) are the means of the replicates from the *in vitro* spacer assay (Figure 3.2g, Table 3.5). Linear regressions (solid lines) and R-values are shown.

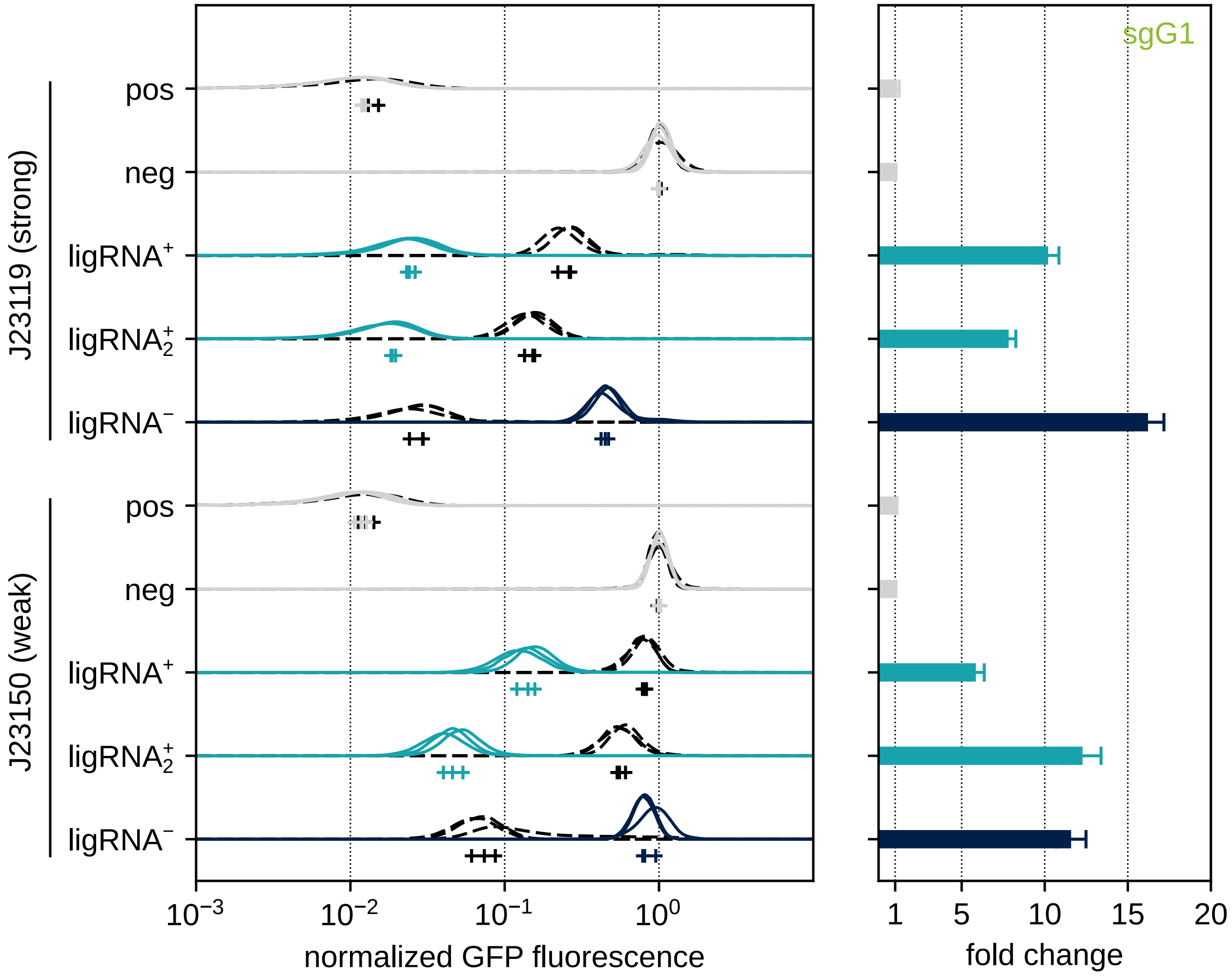

Figure 3.9: ligRNA activity with two promoter strengths. Flow cytometry traces and fold changes for three ligRNAs (ligRNA$^+$ and ligRNA$^-$ discussed in the main text and a third sequence termed ligRNA$^+{}_2$ isolated from the same screen as ligRNA$^+$, Table 3.3) in the context of a strong (J23119) and a weak (J23150) constitutive promoter. GFP fluorescence values for each cell are normalized by both RFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All lines and symbols are described in Figure 3.2. The alternate ligRNA$^+{}_2$ has a larger dynamic range with the weaker promoter, and may be useful for applications where lower concentrations of ligRNA are anticipated. Moreover, in the context of the weak promoter, the fluorescence distributions of ligRNA$^+{}_2$ in the absence of the ligand and the distributions of ligRNA$^-$ in the presence of the ligand (inactive states) are close to the fluorescence distributions of the negative control, which may be useful when one of the desired states is full activation of gene expression.

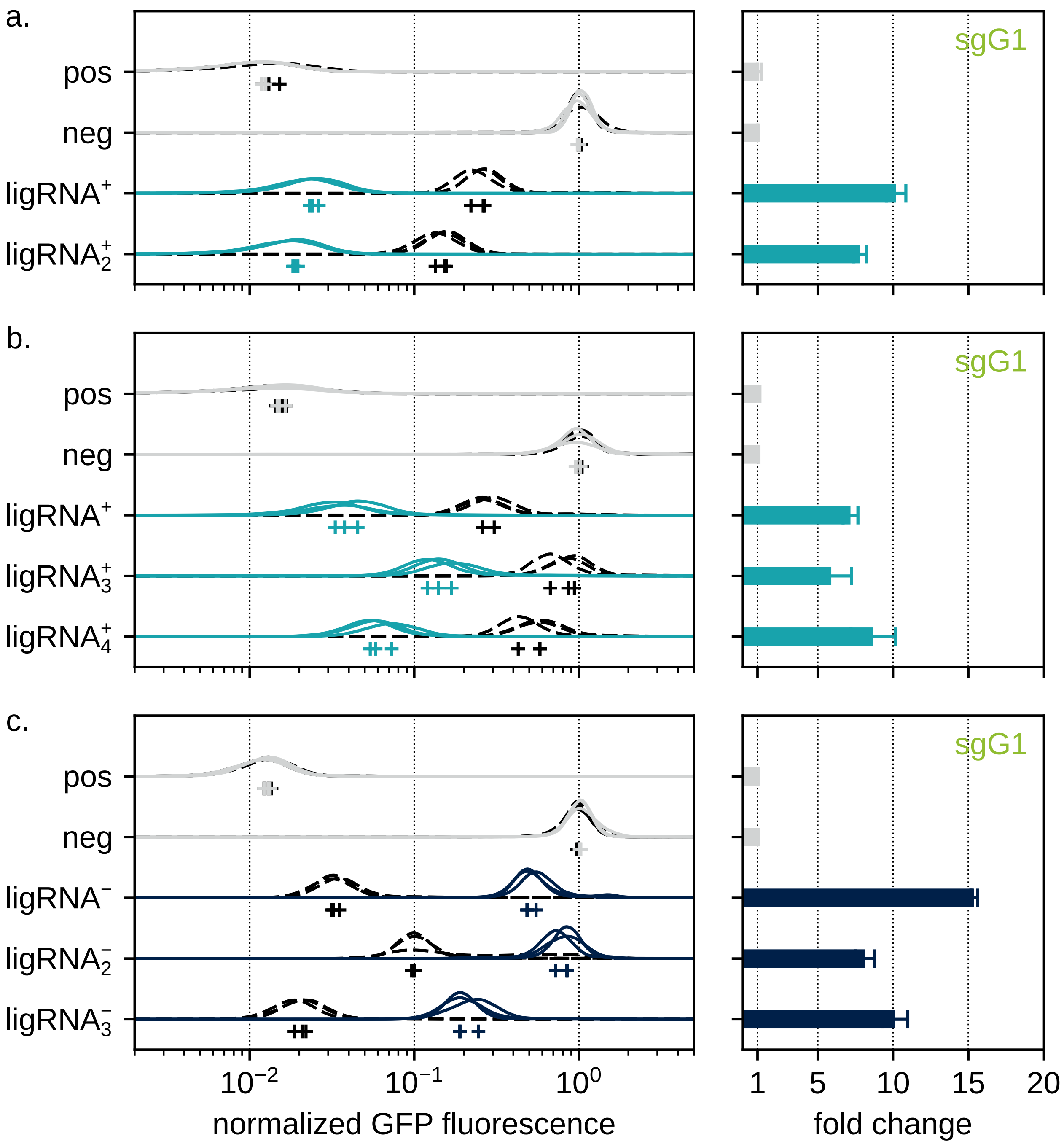

Figure 3.10: ligRNA variants with shifted dynamic ranges. Flow cytometry traces and fold changes for sequence variants of $ligRNA^+$ (a,b) and $ligRNA^-$ (c) that maintain significant sensitivity to theophylline, but shift the dynamic range either towards maximum expression ($ligRNA^+{}_3$ and $ligRNA^+{}_4$ in the absence of the ligand, panel b; $ligRNA^-{}_2$ in the presence of the ligand, panel c) or maximum repression ($ligRNA^+{}_2$ in the presence of the ligand, panel a; $ligRNA^-{}_3$ in the absence of the ligand, panel c). See Table 3.4 for the sequences of these variants. GFP fluorescence values for each cell are normalized by both RFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All lines and symbols are described in Figure 3.2.

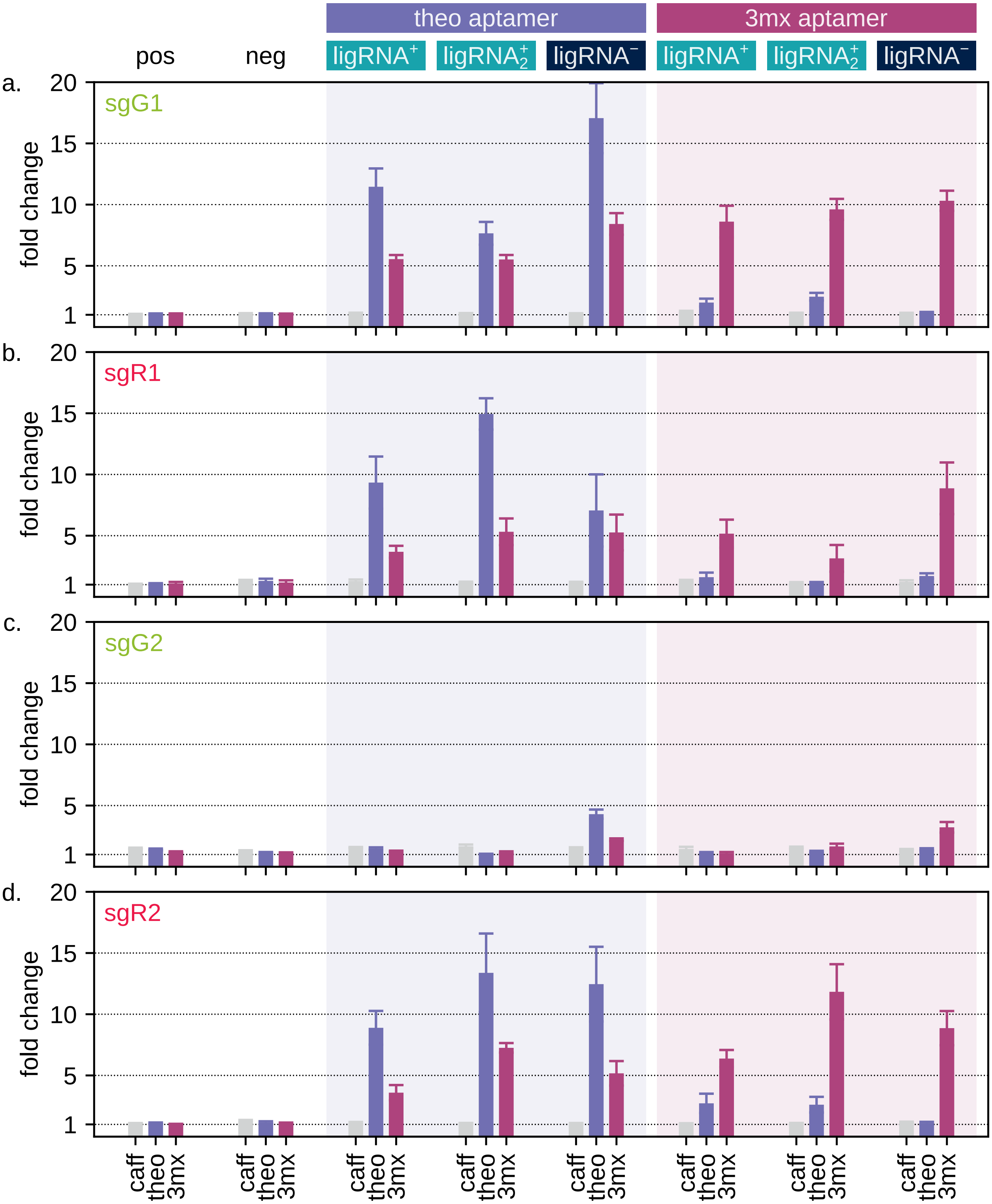
theo aptamer
3mx aptamer
pos
neg
ligRNA+
ligRNA2+
ligRNA−
ligRNA+
ligRNA2+
ligRNA−
a.
sgG1
b.
sgR1
c.
sgG2
d.
sgR2
fold change
20
15
10
5
1
caff
theo
3mx

Figure 3.11: ligRNA function with two different aptamers, four different spacers and three different ligands. We built versions of each ligRNA with both the theophylline aptamer (purple heading) and the 3-methylxanthine aptamer (magenta heading) and then tested them with the three different ligands indicated at the bottom: caffeine (caff, grey bars), theophylline (theo, purple bars), and 3-methylxanthine (3mx, magenta bars). Caffeine is a negative control; it is chemically similar to theophylline and 3-methylxanthine, but is not expected to bind to either aptamer at the concentrations used. The reported fold changes are relative to treatment with no ligand and are calculated from the modes of fluorescence distributions measured by flow cytometry. Panels (a-d) show data for the sgG1, sgR1, sgG2, and sgR2 spacers (Table 3.4), respectively. Note that the theophylline aptamer (purple shading) is expected to be sensitive to both theophylline and 3-methylxanthine, but that the 3-methylxanthine aptamer (magenta shading) is expected to be specific to its ligand 3-methylxanthine with much reduced or no sensitivity to theophylline. The tested ligRNAs behave as expected with the different aptamers and ligands (note that none of the ligRNAs except ligRNA$^-$ showed significant ligand sensitivity with the sgG2 spacer, which we expected based on our prior results shown in Table 3.3). This data was collected by James Lucas.

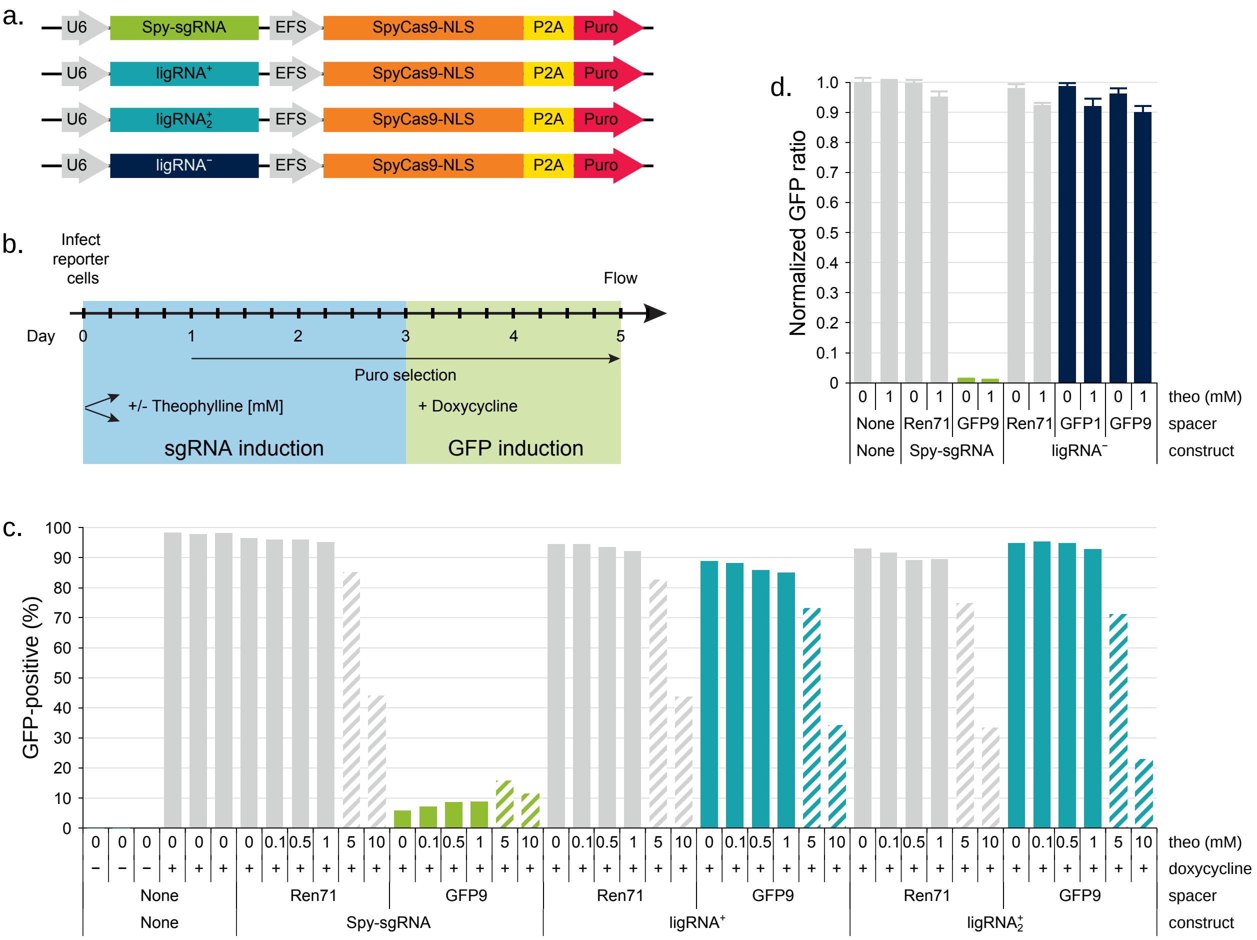

Figure 3.12: Test of ligRNA-mediated target editing in mammalian cell lines. (a) Vector maps of lentiviral constructs expressing Cas9 and either a standard sgRNA as control, or a ligRNA. (b) Schematic of the GFP knockout assay in a HEK293T-based reporter cell line with doxycycline-controlled GFP expression. (c) ligRNA$^+$ does not exhibit theophylline-dependent editing in this assay. Theophylline concentrations from 0.1 to 10 mM were tested. Concentrations above 1 mM caused severe cell death (dashed bars). sgGFP9 targets the GFP of the reporter cell line. sgRen71 is a negative control sgRNA. Note, ligRNA$^+$ resulted in a slight decrease of GFP-positive cells, but this trend began in absence of theophylline. (d) ligRNA$^-$ does not exhibit theophylline-dependent editing in this assay. The slight differences between the 0 mM and 1 mM bars cannot be attributed to ligRNA$^-$, because the Ren71 control has a similar difference; in addition, we would expect editing with ligRNA$^-$ to be inhibited, not activated, by theophylline. This data was collected by Christof Fellman.

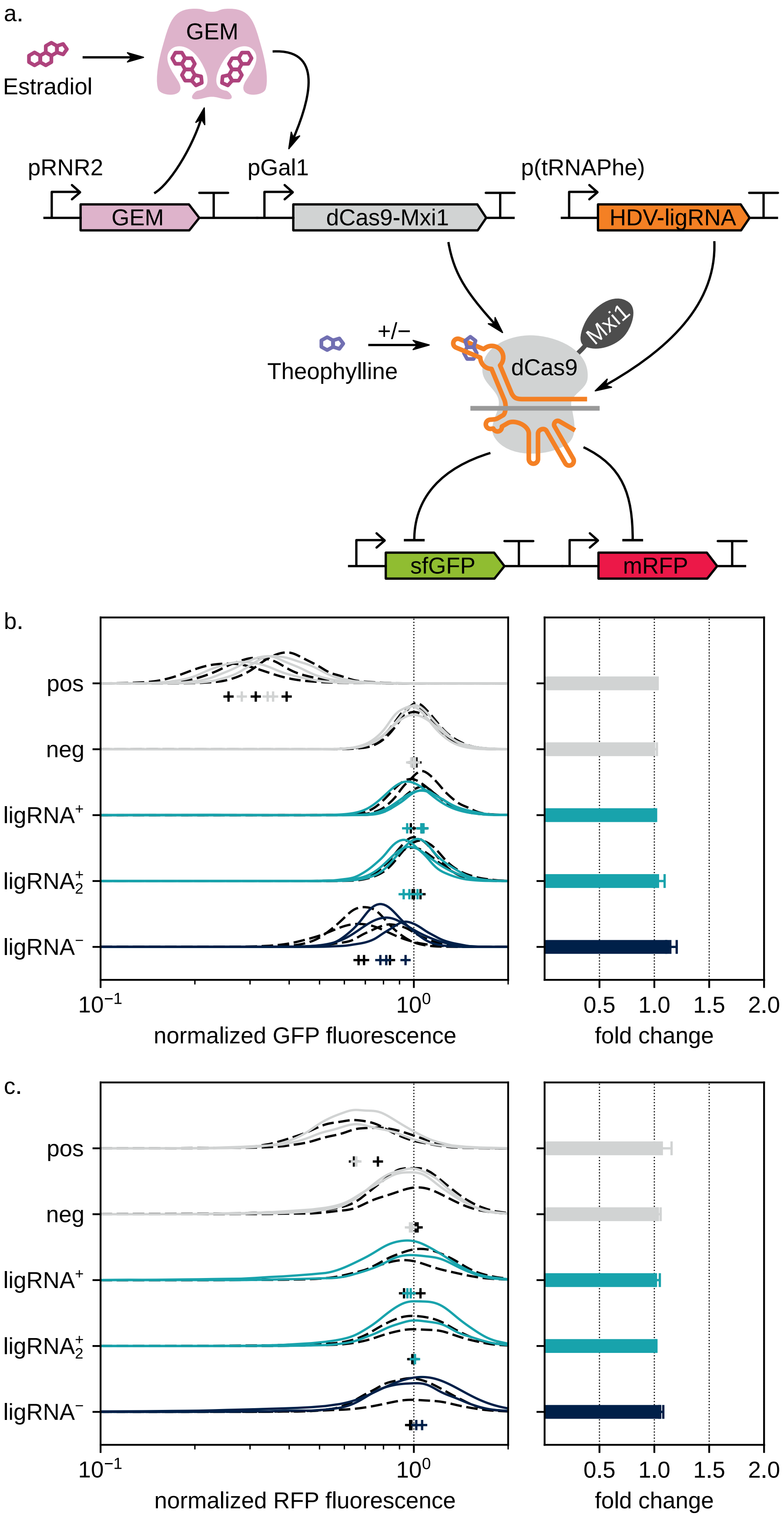
a.
Estradiol
GEM
pRNR2
GEM
pGal1
dCas9-Mxi1
p(tRNAPhe)
HDV-ligRNA
Theophylline
+/−
Mxi1
dCas9
sfGFP
mRFP
b.
pos
neg
ligRNA+
ligRNA2+
ligRNA−
10−1
100
normalized GFP fluorescence
0.5
1.0
1.5
2.0
fold change
c.
pos
neg
ligRNA+
ligRNA2+
ligRNA−
10−1
100
normalized RFP fluorescence
0.5
1.0
1.5
2.0
fold change

Figure 3.13: Test of ligRNA-mediated gene repression in yeast cells. Flow cytometry traces and fold changes for the ligRNAs and positive and negative controls in a CRISPRi assay in S. cerevisiae. (a) Schematic depicting the various constructs in the engineered yeast strains. (b) GFP repression with the sgG2 spacer. ligRNA$^+$ and ligRNA$^+{}_2$ do not exhibit ligand-dependent activity in this assay. ligRNA$^-$ exhibits a weak effect in the expected direction. (c) RFP repression with the sgR2 spacer does not show ligand dependence in this assay. The theophylline concentration in the plus ligand condition was 2.5 mM. The expression of dCas9-Mxi1 was induced with 125nM estradiol. Fluorescence values for each cell are normalized by side scatter (SSC) and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All lines and symbols are described in Figure 3.2. This data was collected by Andrew Ng and Ben Heineike.

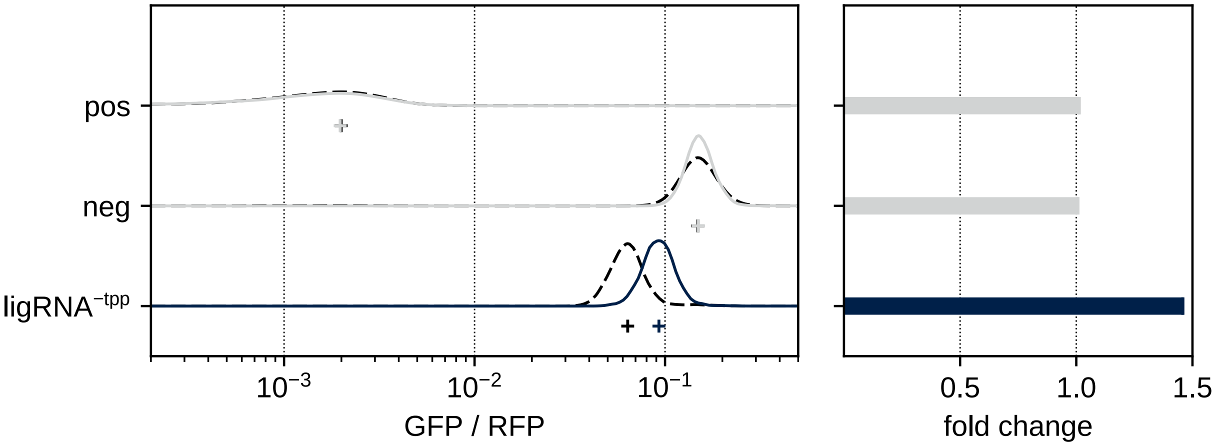

Figure 3.14: ligRNA$^-$ exhibits a small response to thiamine. GFP fluorescence values for each cell are normalized by both RFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All lines and symbols are described in Figure 3.2.

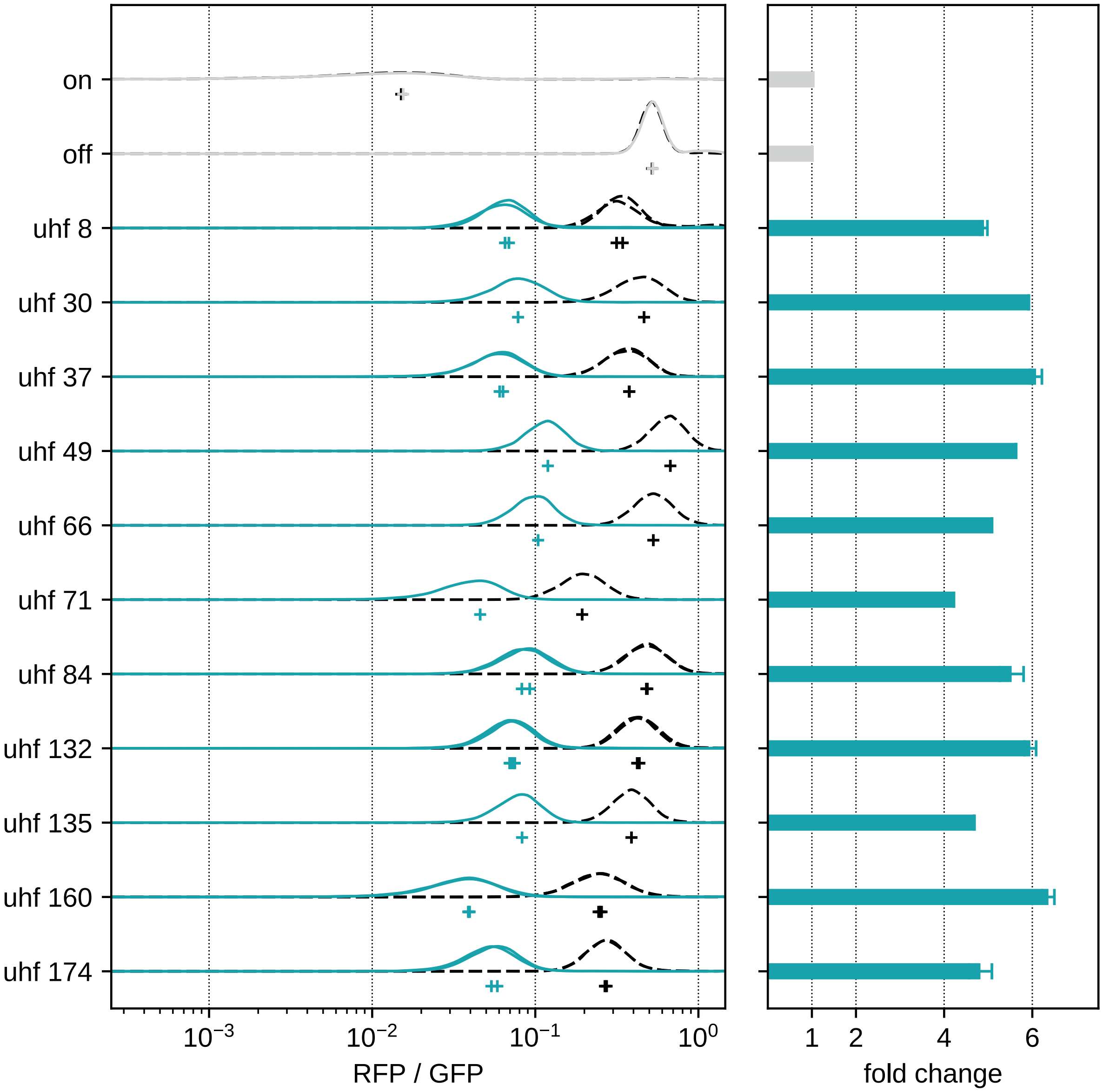

Figure 3.15: Flow cytometry traces for all 11 unique TPP-sensitive sgRNAs. RFP fluorescence values for each cell are normalized by both GFP fluorescence for that cell and the modes of the un-repressed control populations (i.e. apo and holo) measured for that replicate. All lines and symbols are described in Figure 3.2.

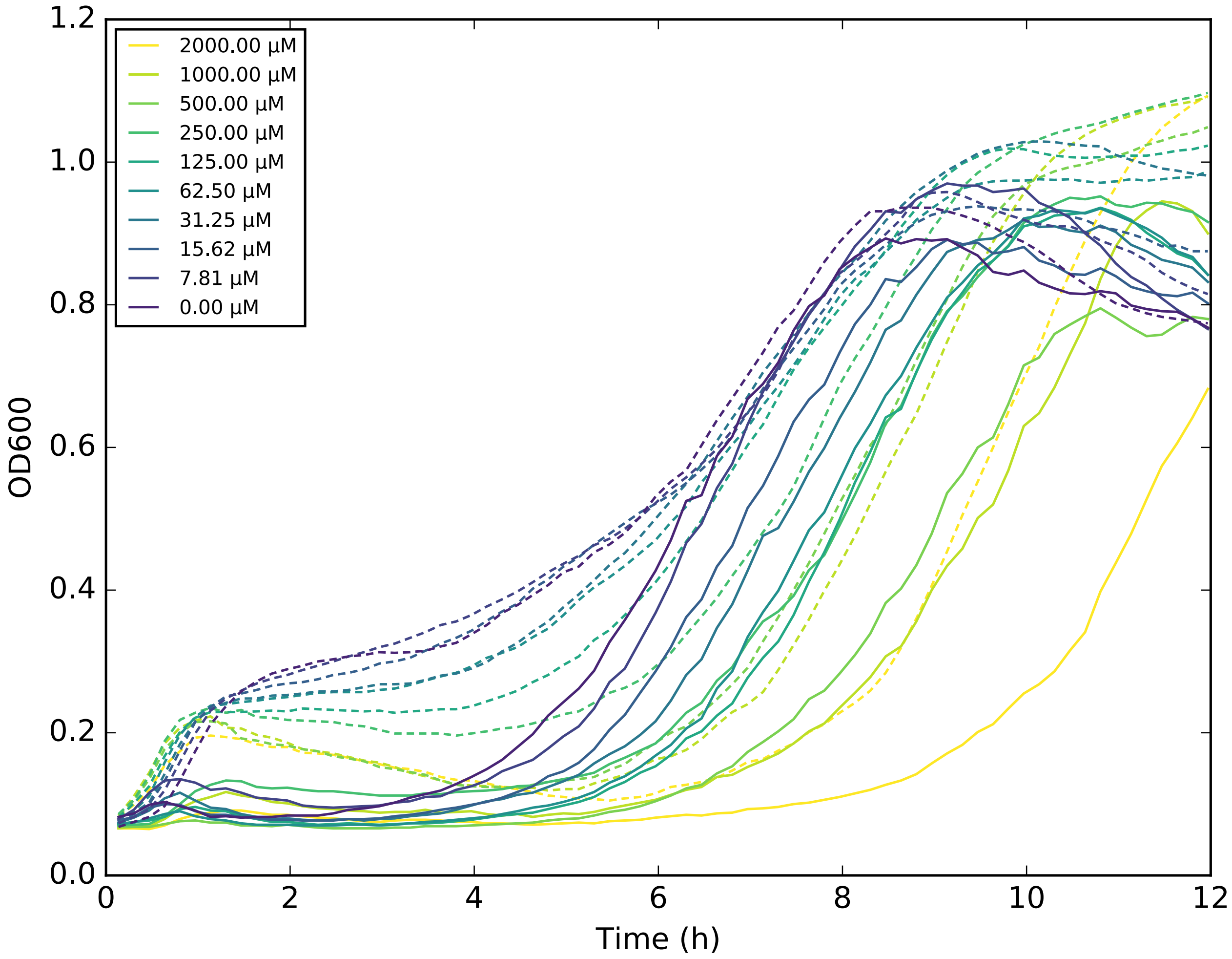

Figure 3.16: Exogenous adenine increases lag time at concentrations as low as 8 µM. Growth curves were recorded for the bacterial CRISPRi strain growing in 200 µL adenine media (see Methods) at 37°C for 12h. Dashed and solid lines represent two biological replicates.

3.6 References

[1] Martin Jinek et al. “A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity”. en. In: *Science* 337.6096 (Aug. 2012), pp. 816–821. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1225829. URL: http://science.sciencemag.org/content/337/6096/816 (visited on 05/30/2018) (cit. on pp. 97, 121).

[2] Lei S. Qi et al. “Repurposing CRISPR as an RNA-Guided Platform for Sequence-Specific Control of Gene Expression”. In: *Cell* 152.5 (Feb. 2013), pp. 1173–1183. DOI: 10.1016/j.cell.2013.02.022 (cit. on pp. 97, 109, 123).

[3] Luke A. Gilbert et al. “CRISPR-Mediated Modular RNA-Guided Regulation of Transcription in Eukaryotes”. In: *Cell* 154.2 (July 2013), pp. 442–451. ISSN: 0092-8674. DOI: 10.1016/j.cell.2013.06.044. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3770145/ (visited on 10/11/2016) (cit. on pp. 97, 113, 127).

[4] Pablo Perez-Pinera et al. “RNA-guided gene activation by CRISPR-Cas9–based transcription factors”. en. In: *Nature Methods* 10.10 (Oct. 2013), pp. 973–976. ISSN: 1548-7105. DOI: 10.1038/nmeth.2600. URL: https://www.nature.com/articles/nmeth.2600 (visited on 05/30/2018) (cit. on p. 97).

[5] Baohui Chen et al. “Dynamic Imaging of Genomic Loci in Living Human Cells by an Optimized CRISPR/Cas System”. In: *Cell* 155.7 (Dec. 2013), pp. 1479–1491. ISSN: 0092-8674. DOI: 10.1016/j.cell.2013.12.001. URL: http://www.sciencedirect.com/science/article/pii/S0092867413015316 (visited on 07/11/2016) (cit. on pp. 97, 115).

[6] Gregory M. Findlay et al. “Saturation editing of genomic regions by multiplex homology-directed repair”. en. In: *Nature* 513.7516 (Sept. 2014), pp. 120–123. ISSN: 1476-4687. DOI: 10.1038/nature13695. URL: https://www.nature.com/articles/nature13695 (visited on 05/30/2018) (cit. on p. 97).

[7] Isaac B. Hilton et al. “Epigenome editing by a CRISPR-Cas9-based acetyltransferase activates genes from promoters and enhancers”. en. In: *Nature Biotechnology* 33.5 (May 2015), pp. 510–517. ISSN: 1546-1696. DOI: 10.1038/nbt.3199. URL: https://www.nature.com/articles/nbt.3199 (visited on 05/30/2018) (cit. on p. 97).

[8] James K. Nuñez, Lucas B. Harrington, and Jennifer A. Doudna. “Chemical and Biophysical Modulation of Cas9 for Tunable Genome Engineering”. In: *ACS Chemical Biology* 11.3 (Mar. 2016), pp. 681–688. ISSN: 1554-8929. DOI: 10.1021/acschembio.5b01019. URL: https://doi.org/10.1021/acschembio.5b01019 (visited on 05/30/2018) (cit. on p. 97).

[9] Florian Richter et al. “Switchable Cas9”. In: *Curr. Opin. Biotechnol.* 48 (Dec. 2017), pp. 119–126. DOI: 10.1016/j.copbio.2017.03.025 (cit. on p. 97).

[10] Kevin M. Davis et al. “Small molecule–triggered Cas9 protein with improved genome-editing specificity”. en. In: *Nature Chemical Biology* 11.5 (May 2015), pp. 316–318. ISSN: 1552-4469. DOI: 10.1038/nchembio.1793. URL: https://www.nature.com/articles/nchembio.1793 (visited on 05/30/2018) (cit. on p. 97).

[11] Yuchen Liu et al. “Directing cellular information flow via CRISPR signal conductors”. In: *Nat. Methods* 13 (Sept. 2016), pp. 938–944. DOI: 10.1038/nmeth.3994 (cit. on p. 97).

[12] Basudeb Maji et al. “Multidimensional chemical control of CRISPR-Cas9”. In: *Nat. Chem. Biol.* 13 (Jan. 2017), pp. 9–11. DOI: 10.1038/nchembio.2224 (cit. on p. 97).

[13] Nguyen et al. “Ligand-binding domains of nuclear receptors facilitate tight control of split CRISPR activity”. In: *Nat. Commun.* 7 (July 2016), p. 12009. DOI: 10.1038/ncomms12009 (cit. on p. 97).

[14] Benjamin L. Oakes et al. “Profiling of engineering hotspots identifies an allosteric CRISPR-Cas9 switch”. In: *Nat. Biotechnol.* 34.6 (June 2016), pp. 646–651. DOI: 10.1038/nbt.3528 (cit. on pp. 97, 100).

[15] Bernd Zetsche, Sara E Volz, and Feng Zhang. “A split-Cas9 architecture for inducible genome editing and transcription modulation”. In: *Nat. Biotechnol.* 33 (Feb. 2015), pp. 139–142. DOI: 10.1038/nbt.3149 (cit. on p. 97).

[16] Yuchen Gao et al. “Complex transcriptional modulation with orthogonal and inducible dCas9 regulators”. en. In: *Nature Methods* 13.12 (Dec. 2016), pp. 1043–1049. ISSN: 1548-7105. DOI: 10.1038/nmeth.4042. URL: https://www.nature.com/articles/nmeth.4042 (visited on 05/30/2018) (cit. on p. 97).

[17] James Hemphill et al. “Optical Control of CRISPR/Cas9 Gene Editing”. In: *Journal of the American Chemical Society* 137.17 (May 2015), pp. 5642–5645. DOI: 10.1021/ja512664v (cit. on p. 97).

[18] Yuta Nihongaki et al. “Photoactivatable CRISPR-Cas9 for optogenetic genome editing”. In: *Nat. Biotechnol.* 33 (June 2015), pp. 755–760. DOI: 10.1038/nbt.3245 (cit. on p. 97).

[19] Lauren R. Polstein and Charles A. Gersbach. “A light-inducible CRISPR-Cas9 system for control of endogenous gene activation”. en. In: *Nature Chemical Biology* 11.3 (Mar. 2015), pp. 198–200. ISSN: 1552-4469. DOI: 10.1038/nchembio.1753. URL: https://www.nature.com/articles/nchembio.1753 (visited on 05/30/2018) (cit. on p. 97).

[20] Zehua Bao et al. “Orthogonal Genetic Regulation in Human Cells Using Chemically Induced CRISPR/Cas9 Activators”. In: *ACS Synthetic Biology* 6.4 (Apr. 2017), pp. 686–693. DOI: 10.1021/acssynbio.6b00313. URL: https://doi.org/10.1021/acssynbio.6b00313 (visited on 05/30/2018) (cit. on p. 97).

[21] Kevin M. Esvelt et al. “Orthogonal Cas9 proteins for RNA-guided gene regulation and editing”. en. In: *Nature Methods* 10.11 (Nov. 2013), pp. 1116–1121. ISSN: 1548-7105. DOI: 10.1038/nmeth.2681. URL: https://www.nature.com/articles/nmeth.2681 (visited on 05/30/2018) (cit. on p. 97).

[22] Yuchen Liu et al. “Directing cellular information flow via CRISPR signal conductors”. en. In: *Nature Methods* advance online publication (Sept. 2016). ISSN: 1548-7091. DOI: 10.1038/nmeth.3994. URL: http://www.nature.com/nmeth/journal/vaop/ncurrent/full/nmeth.3994.html (visited on 10/24/2016) (cit. on p. 97).

[23] Weixin Tang, Johnny H. Hu, and David R. Liu. “Aptazyme-embedded guide RNAs enable ligand-responsive genome editing and transcriptional activation”. In: *Nat. Commun.* 8 (June 2017), p. 15939. DOI: 10.1038/ncomms15939 (cit. on p. 97).

[24] Quentin R. V. Ferry, Radostina Lyutova, and Tudor A. Fulga. “Rational design of inducible CRISPR guide RNAs for *de novo* assembly of transcriptional programs”. en. In: *Nature Communications* 8 (Mar. 2017), ncomms14633. ISSN: 2041-1723. DOI: 10.1038/ncomms14633.

URL: https://www.nature.com/articles/ncomms14633 (visited on 11/14/2017) (cit. on p. 97).

[25] Young Je Lee et al. “Programmable control of bacterial gene expression with the combined CRISPR and antisense RNA system”. In: *Nucleic Acids Research* 44.5 (Mar. 2016), pp. 2462–2473. ISSN: 0305-1048. DOI: 10.1093/nar/gkw056. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4797300/ (visited on 05/30/2018) (cit. on p. 97).

[26] Alexandra E. Briner et al. “Guide RNA functional modules direct Cas9 activity and orthogonality”. In: *Mol. Cell* 56.2 (Oct. 2014), pp. 333–339. DOI: 10.1016/j.molcel.2014.09.019 (cit. on pp. 98, 99, 102, 108, 112).

[27] Beatrix Suess et al. “A theophylline responsive riboswitch based on helix slipping controls gene expression in vivo”. In: *Nucleic Acids Research* 32.4 (2004), pp. 1610–1614. ISSN: 0305-1048. DOI: 10.1093/nar/gkh321. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC390306/ (visited on 02/01/2016) (cit. on p. 98).

[28] Ying Dang et al. “Optimizing sgRNA structure to improve CRISPR-Cas9 knockout efficiency”. In: *Genome Biology* 16 (2015), p. 280. ISSN: 1474-760X. DOI: 10.1186/s13059-015-0846-3. URL: http://dx.doi.org/10.1186/s13059-015-0846-3 (visited on 07/11/2016) (cit. on pp. 102, 112).

[29] Ronny Lorenz et al. “ViennaRNA Package 2.0”. In: *Algorithms Mol. Biol.* 6.26 (Nov. 2011). DOI: 10.1186/1748-7188-6-26 (cit. on pp. 100, 103).

[30] Hiroshi Nishimasu et al. “Crystal structure of Cas9 in complex with guide RNA and target DNA”. In: *Cell* 156.5 (Feb. 2014), pp. 935–949. DOI: 10.1016/j.cell.2014.02.001 (cit. on p. 102).

[31] Robert D. Jenison et al. “High-resolution molecular discrimination by RNA”. In: *Science* 263 (Mar. 1994), pp. 1425–1429. DOI: 10.1126/science.7510417 (cit. on pp. 103, 112).

[32] Chase L Beisel et al. “Model-guided design of ligand-regulated RNAi for programmable control of gene expression”. In: *Molecular Systems Biology* 4 (Oct. 2008), p. 224. ISSN: 1744-

4292. DOI: 10.1038/msb.2008.62. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2583087/ (visited on 03/10/2018) (cit. on p. 103).

[33] Joe C. Liang and Christina D. Smolke. “Rational design and tuning of ribozyme-based devices”. In: *Ribozymes: Methods and Protocols*. Ed. by Jörg S. Hartig. Methods Mol. Biol. Totowa, NJ: Humana Press, 2012. Chap. 27, pp. 439–454. DOI: 10.1007/978-1-61779-545-9_27 (cit. on p. 103).

[34] Garrett A Soukup, Gail A. M Emilsson, and Ronald R Breaker. “Altering molecular recognition of RNA aptamers by allosteric selection1”. In: *Journal of Molecular Biology* 298.4 (May 2000), pp. 623–632. ISSN: 0022-2836. DOI: 10.1006/jmbi.2000.3704. URL: http://www.sciencedirect.com/science/article/pii/S0022283600937045 (visited on 01/26/2016) (cit. on p. 105).

[35] Michael T. Laub et al. “Global Analysis of the Genetic Network Controlling a Bacterial Cell Cycle”. en. In: *Science* 290.5499 (Dec. 2000), pp. 2144–2148. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.290.5499.2144. URL: http://science.sciencemag.org/content/290/5499/2144 (visited on 05/30/2018) (cit. on p. 105).

[36] Jason M. Peters et al. “A comprehensive, CRISPR-based functional analysis of essential genes in bacteria”. In: *Cell* 165.6 (June 2016), pp. 1493–1506. DOI: 10.1016/j.cell.2016.05.003 (cit. on p. 105).

[37] Markus Wieland et al. “Artificial Ribozyme Switches Containing Natural Riboswitch Aptamer Domains”. en. In: *Angewandte Chemie International Edition* 48.15 (Mar. 2009), pp. 2715–2718. ISSN: 1521-3773. DOI: 10.1002/anie.200805311. URL: http://onlinelibrary.wiley.com/doi/10.1002/anie.200805311/abstract (visited on 01/29/2016) (cit. on p. 106).

[38] Wade Winkler, Ali Nahvi, and Ronald R. Breaker. “Thiamine derivatives bind messenger RNAs directly to regulate bacterial gene expression”. en. In: *Nature* 419.6910 (Oct. 2002), pp. 952–956. ISSN: 0028-0836. DOI: 10.1038/nature01145. URL: http://www.nature.com/nature/journal/v419/n6910/full/nature01145.html (visited on 10/12/2016) (cit. on p. 106).

[39] Ghislain Schyns et al. “Isolation and Characterization of New Thiamine-Deregulated Mutants of Bacillus subtilis”. en. In: *Journal of Bacteriology* 187.23 (Dec. 2005), pp. 8127–8136. ISSN: 0021-9193, 1098-5530. DOI: 10.1128/JB.187.23.8127-8136.2005. URL: http://jb.asm.org/content/187/23/8127 (visited on 05/26/2018) (cit. on p. 106).

[40] Neil Dixon et al. “Reengineering orthogonally selective riboswitches”. en. In: *Proceedings of the National Academy of Sciences* 107.7 (Feb. 2010), pp. 2830–2835. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.0911209107. URL: http://www.pnas.org/content/107/7/2830 (visited on 02/01/2016) (cit. on pp. 106, 107).

[41] R. A. Levine and M. W. Taylor. “Mechanism of adenine toxicity in Escherichia coli.” en. In: *Journal of Bacteriology* 149.3 (Mar. 1982), pp. 923–930. ISSN: 0021-9193, 1098-5530. URL: http://jb.asm.org/content/149/3/923 (visited on 02/21/2017) (cit. on p. 106).

[42] Jenny L. Baker et al. “Widespread Genetic Switches and Toxicity Resistance Proteins for Fluoride”. In: *Science (New York, N.Y.)* 335.6065 (Jan. 2012), pp. 233–235. ISSN: 0036-8075. DOI: 10.1126/science.1215063. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4140402/ (visited on 05/24/2018) (cit. on p. 107).

[43] Aiming Ren, Kanagalaghatta R. Rajashankar, and Dinshaw J. Patel. “Fluoride ion encapsulation by Mg2+ and phosphates in a fluoride riboswitch”. In: *Nature* 486.7401 (May 2012), pp. 85–89. ISSN: 0028-0836. DOI: 10.1038/nature11152. URL: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3744881/ (visited on 10/11/2016) (cit. on p. 107).

[44] Haili Ma et al. “Effects of fluoride on bacterial growth and its gene/protein expression”. In: *Chemosphere* 100 (Apr. 2014), pp. 190–193. ISSN: 0045-6535. DOI: 10.1016/j.chemosphere.2013.11.032. URL: http://www.sciencedirect.com/science/article/pii/S0045653513016226 (visited on 05/24/2018) (cit. on p. 107).

[45] Tim Wang et al. “Genetic Screens in Human Cells Using the CRISPR-Cas9 System”. en. In: *Science* 343.6166 (Jan. 2014), pp. 80–84. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1246981. URL: http://science.sciencemag.org/content/343/6166/80 (visited on 05/24/2018) (cit. on p. 107).

[46] Hiroko Koike-Yusa et al. “Genome-wide recessive genetic screening in mammalian cells with a lentiviral CRISPR-guide RNA library”. en. In: *Nature Biotechnology* 32.3 (Mar. 2014), pp. 267–273. ISSN: 1546-1696. DOI: 10.1038/nbt.2800. URL: https://www.nature.com/articles/nbt.2800 (visited on 05/24/2018) (cit. on p. 107).

[47] Ophir Shalem et al. “Genome-Scale CRISPR-Cas9 Knockout Screening in Human Cells”. en. In: *Science* 343.6166 (Jan. 2014), pp. 84–87. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.1247005. URL: http://science.sciencemag.org/content/343/6166/84 (visited on 05/24/2018) (cit. on p. 107).

[48] Johannes Schindelin et al. “Fiji: an open-source platform for biological-image analysis”. en. In: *Nature Methods* 9.7 (July 2012), pp. 676–682. ISSN: 1548-7105. DOI: 10.1038/nmeth.2019. URL: https://www.nature.com/articles/nmeth.2019 (visited on 05/30/2018) (cit. on p. 109).

[49] Jean-Denis Pédelacq et al. “Engineering and characterization of a superfolder green fluorescent protein”. In: *Nat. Biotechnol.* 24.1 (Dec. 2006), pp. 79–88. DOI: 10.1038/nbt1172 (cit. on p. 109).

[50] Robert E. Campbell et al. “A monomeric red fluorescent protein”. In: *Proc. Natl. Acad. Sci. U.S.A* 99.12 (June 2002), pp. 7877–7882. DOI: 10.1073/pnas.082243699 (cit. on p. 109).

[51] Mira I. Pronobis, Natalie Deuitch, and Mark Peifer. “The Miraprep: A Protocol that Uses a Miniprep Kit and Provides Maxiprep Yields”. In: *PLoS ONE* 11.8 (Aug. 2016). DOI: 10.1371/journal.pone.0160509 (cit. on p. 110).

[52] John G. Doench et al. “Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9.” In: *Nat. Biotechnol.* 34.2 (Jan. 2016), pp. 184–191. DOI: 10.1038/nbt.3437 (cit. on p. 112).

[53] Michael E. Lee et al. “A Highly Characterized Yeast Toolkit for Modular, Multipart Assembly”. In: *ACS Synthetic Biology* 4.9 (Sept. 2015), pp. 975–986. DOI: 10.1021/sb500366v. URL: https://doi.org/10.1021/sb500366v (visited on 05/30/2018) (cit. on pp. 113, 116).

[54] William C. DeLoache et al. “An enzyme-coupled biosensor enables (*S*)-reticuline production in yeast from glucose”. en. In: *Nature Chemical Biology* 11.7 (July 2015), pp. 465–471.

ISSN: 1552-4469. DOI: 10.1038/nchembio.1816. URL: https://www.nature.com/articles/nchembio.1816 (visited on 05/30/2018) (cit. on p. 113).

[55] Andrés Aranda-Díaz et al. “Robust Synthetic Circuits for Two-Dimensional Control of Gene Expression in Yeast”. In: *ACS Synthetic Biology* 6.3 (Mar. 2017), pp. 545–554. DOI: 10.1021/acssynbio.6b00251. URL: https://doi.org/10.1021/acssynbio.6b00251 (visited on 05/30/2018) (cit. on p. 113).

[56] Owen W. Ryan and Jamie H. D. Cate. “Multiplex Engineering of Industrial Yeast Genomes Using CRISPRm”. In: *Methods in Enzymology*. Ed. by Jennifer A. Doudna and Erik J. Sontheimer. Vol. 546. The Use of CRISPR/Cas9, ZFNs, and TALENs in Generating Site-Specific Genome Alterations. Academic Press, Jan. 2014, pp. 473–489. DOI: 10.1016/B978-0-12-801185-0.00023-4. URL: http://www.sciencedirect.com/science/article/pii/B9780128011850000234 (visited on 05/30/2018) (cit. on p. 113).

[57] Christof Fellmann et al. “An Optimized microRNA Backbone for Effective Single-Copy RNAi”. In: *Cell Reports* 5.6 (Dec. 2013), pp. 1704–1713. ISSN: 2211-1247. DOI: 10.1016/j.celrep.2013.11.020. URL: http://www.sciencedirect.com/science/article/pii/S2211124713006876 (visited on 05/30/2018) (cit. on p. 114).

[58] Neville E. Sanjana, Ophir Shalem, and Feng Zhang. “Improved vectors and genome-wide libraries for CRISPR screening”. In: *Nature methods* 11.8 (Aug. 2014), pp. 783–784. ISSN: 1548-7091. DOI: 10.1038/nmeth.3047. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4486245/ (visited on 05/30/2018) (cit. on p. 114).

[59] R. Daniel Gietz and Robert H. Schiestl. “Large-scale high-efficiency yeast transformation using the LiAc/SS carrier DNA/PEG method”. en. In: *Nature Protocols* 2.1 (Jan. 2007), pp. 38–41. ISSN: 1754-2189. DOI: 10.1038/nprot.2007.15. URL: http://www.nature.com/nprot/journal/v2/n1/full/nprot.2007.15.html (visited on 03/09/2017) (cit. on p. 115).

Publishing Agreement

It is the policy of the University to encourage the distribution of all theses, dissertations, and manuscripts. Copies of all UCSF theses, dissertations, and manuscripts will be routed to the library via the Graduate Division. The library will make all theses, dissertations, and manuscripts accessible to the public and will preserve these to the best of their abilities, in perpetuity.

Please sign the following statement:

I hereby grant permission to the Graduate Division of the University of California, San Francisco to release copies of my thesis, dissertation, or manuscript to the Campus Library to provide access and preservation, in whole or in part, in perpetuity.

______________________________ 6/8/18

Author Signature Date