

---

# Unlocking hidden biomolecular conformational landscapes in diffusion models at inference time

---

**Daniel D. Richman\***  
Stanford University  
ddrichma@stanford.edu

**Jessica Karaguesian\***  
Stanford University  
jkara@stanford.edu

**Carl-Mikael Suomivuori**  
Stanford University  
carl-mikael.suomivuori@yale.edu

**Ron O. Dror**  
Stanford University  
rondror@cs.stanford.edu

## 1 Supplementary information

In the pages below we provide additional figures and code for key sections of our paper. We further refer the reader interested in code to the Boltz (<https://github.com/jwohlwend/boltz/tree/main>) and BioEmu (<https://github.com/microsoft/bioemu/tree/main>) repositories, as well as the reference twisted diffusion implementation at [https://github.com/blt2114/twisted\\_diffusion\\_sampler](https://github.com/blt2114/twisted_diffusion_sampler) and to the reference MBAR implementation at <https://pymbar.readthedocs.io/en/master/index.html>.

## 2 Datasets

- **Domain motion:** 38 proteins consisting of the non-overlapping set of 22 domain motion proteins curated by [3] and 23 open-closed (OC23) conformation proteins curated by [1].
- **Membrane transporters:** 15 proteins from the transporter protein set (TP16) curated by [1]. One protein, SPF1, was excluded for compute considerations due to its long sequence length.
- **Cryptic pockets:** 33 proteins that undergo cryptic pocket formation curated by [3]. RMSD metrics were computed using the cryptic pocket region defined by [3]. One protein, Q16539, was excluded because the region definitions were not available.
- **Fold switching:** 15 proteins that exhibit fold switching were randomly selected from the set of proteins curated by [5] and used to benchmark other conformational prediction methods including CF-random ([2]).

## 3 Computational resources and timing

Compute for this work was run primarily on a cluster of 8 A40 GPUs. We typically generate 800 samples per protein for ConforMixRMSD, although motions of interest are typically visible more quickly. A rapid scan with a small number of samples per RMSD finishes in minutes.

We benchmarked wall clock sampling speeds for three proteins on a single A40 GPU without twisting (i.e., default sampling) and with twisting (i.e., ConforMix), for different numbers of simultaneous samples/particles. Because ConforMix modifies only the diffusion module and not the encoder, we separately time these modules. Total Time = encoder + decoder together. The penalty for running

---

\*These authors contributed equally to this work.

twisted diffusion at every timestep of the decoder is 2-3x wallclock speed. We believe our results demonstrate that the generated samples are substantially more informative than those from default sampling.

There is flexibility to tune performance by scheduling guidance and/or resampling only at certain timesteps of the decoder. We have not thoroughly explored the tradeoffs involved.

Table S1: Runtime analysis of ConforMix with Boltz

Protein	# Residues	# Atoms	# Simult. Samples	Diffusion Time (s)		Total Time (s)
				default	ConforMix	default
P0DP23	148	1184	1	$6.9 \pm 0.2$	-	$9.5 \pm 0.1$
			2	-	$28.1 \pm 1.3$	-
			4	$9.7 \pm 0.0$	$31.6 \pm 0.7$	$12.3 \pm 0.0$
			16	$31.7 \pm 0.1$	$71.0 \pm 0.4$	$34.3 \pm 0.1$
P37487	309	2400	1	$7.2 \pm 0.2$	-	$18.7 \pm 0.2$
			2	-	$30.7 \pm 0.7$	-
			4	$20.0 \pm 0.2$	$44.4 \pm 0.2$	$31.5 \pm 0.2$
			16	$70.7 \pm 0.1$	$143.9 \pm 0.6$	$82.1 \pm 0.1$
B0F0C5	497	4000	1	$11.5 \pm 0.0$	-	$47.2 \pm 0.0$
			2	-	$48.6 \pm 0.1$	-
			4	$37.4 \pm 0.2$	$86.9 \pm 0.7$	$73.1 \pm 0.2$
			16	$134.6 \pm 0.1$	$258.9 \pm 0.3$	$170.4 \pm 0.1$

Runtime speed of Boltz and ConforMix-Boltz for three protein systems. Columns: diffusion only and encoder + diffusion for three protein systems, in seconds. # Simultaneous Samples is the number of samples moving through diffusion, either independently of each other (default sampling) or with guidance and resampling (ConforMix). Standard deviation is shown over 5 runs.

## 4 MBAR

In this section we outline the Multistate Bennett Acceptance Ratio algorithm for free energy estimation introduced in Section 3.3. The input to MBAR is samples generated from ConforMix, i.e. collected from a series of conditional probability distributions  $p_j(\mathbf{x})$ , which are distinct from the unconditional distribution  $p(\mathbf{x})$ . For notational convenience, we drop  $s$  in the remainder of this section, and we will write  $p_j(x) = p(x|y_j)$ , where  $y_j$  is the condition we impose.

Suppose we have a set of distributions  $\{p(x|y_1), \dots, p(x|y_J)\}$  and from each  $p(x|y_j)$  we draw i.i.d. a set of  $n_j$  samples  $\{x_i^{(j)}\}$ . The key principle of MBAR is to take advantage of overlap between the distributions. Although a given sample  $x_i^{(j)}$  was generated from its distribution  $p(x|y_j)$ , it is equally correct to model the entire set of samples  $\{x_i^{(j)}\}$  as generated from the mixture distribution  $p_{mix}(x) = \sum_{j \in \{1, \dots, J\}} q(j) p(x|y_j)$ , where we define  $q(j) = \frac{n_j}{\sum_l n_l}$  to be the fraction of samples collected at  $j$ . To estimate the partition functions, we now write the importance sampling estimate for an arbitrary observable  $O(x)$  in an arbitrary ensemble  $p(x|y_j)$ , given samples from the mixture  $p_{mix}(x)$ :

$$\int O(x) p(x|y_j) dx = \int O(x) \frac{p_{mix}(x)}{p_{mix}(x)} p(x|y_j) dx \approx \left\langle O(x) \frac{p(x|y_j)}{p_{mix}(x)} \right\rangle_{mix} \quad (1)$$

Let us choose  $O(x) = 1$  for all  $x$ . Then we have

$$1 = \left\langle \frac{p(x|y_j)}{p_{mix}(x)} \right\rangle_{mix} = \left\langle \frac{p(x|y_j)}{\sum_{r \in \{1, \dots, J\}} q(r) p(x|y_r)} \right\rangle_{mix} \quad (2)$$

Turning to the conditional distributions  $p_j(x)$  sampled by ConforMix, we previously defined the twisting potential  $y_j(x) = \exp(-k_j(g_j(x) - g_j^{(0)}))/Z_j$ , where  $g_j(x)$  is an observable. We then have

$$p(x|y_j) = \frac{p(x)y_j(x)}{p(y_j)} = \frac{p(x) \exp(-k_j(g_j(x) - g_j^{(0)}))/Z_j}{p(y_j)}. \quad (3)$$

Substituting this expression into the numerator and denominator of (2) and rearranging, we obtain

$$p(y_j) \approx \left\langle \frac{\exp(-k_j(g_j(x) - g_j^{(0)}))/Z_j}{\sum_{r \in \{1, \dots, J\}} p(r) \left( \frac{\exp(-k_r(g_r(x) - g_r^{(0)}))}{Z_r p(y_r)} \right)} \right\rangle_{mix}. \quad (4)$$

Crucially,  $p(x)$  cancels, meaning we do not need to evaluate the absolute probability of a given sample under the diffusion model. Finally, observe that the unconditional distribution  $p(x)$ , the target of our reweighting, can also be written as  $p(x|y_0)$  where  $k_0, g_0, g_0^{(0)} = 0$ . Thus, for samples collected from  $J$  biased distributions, we form a system of  $J + 1$  equations in  $J + 1$  unknowns by writing Equation 4 for each  $p(x|y_j)$  plus  $p(x|y_0) = p(x)$ . This system can be solved numerically to obtain the target partition functions. In our use case,  $J$  is typically on the order of 10 to 100, although MBAR can accommodate larger sets if necessary.

## 5 Benchmarking methods for ConforMixRMSD

In Section 4.1 we compare ConforMixRMSD performance to default Boltz sampling and to three other methods: AFCluster, AFsample2, and CF-random. To allow for an apples-to-apples comparison, we reimplement the key steps of each of these methods to produce inputs for Boltz.

For AFsample2, we use the recommended settings of 1000 randomly generated input MSAs with the column randomization fraction set to 0.15. The published CF-random protocol modifies the `-max-seq` and `-max-extra-seq` inputs to AlphaFold 3, but Boltz lacks the `-max-extra-seq` input, so the MSA modifications we make are equivalent to only the first option. We use the suggested settings of `-max-seq = 1, 2, 4, 8, 16, 32, 64, 128, 256, 512` with 50 randomly sampled MSAs in each condition, for a total of 500 CF-random input MSAs per protein system. We run AFCluster with default settings; the number of clusters, and therefore the number of sampled MSAs per system, varies per system depending on the initial MSA.

## 6 Outliers

To assess the structural validity of the generated samples, we examined outliers within the generated samples from the domain motion protein set using the Boltz implementation.

Table S2: Outlier fractions for structural quality metrics by method

	Bond Lengths	Bond Angles	Ramachandran Angles
Default Boltz sampling	$2.96 \times 10^{-3}$	$2.14 \times 10^{-4}$	$6.63 \times 10^{-4}$
AFCluster-Boltz	$5.60 \times 10^{-3}$	$6.02 \times 10^{-4}$	$1.51 \times 10^{-3}$
AFsample2-Boltz	$1.08 \times 10^{-2}$	$2.94 \times 10^{-4}$	$2.13 \times 10^{-3}$
CF-random-Boltz	$1.08 \times 10^{-2}$	$2.94 \times 10^{-4}$	$2.13 \times 10^{-3}$
ConforMixRMSD-Boltz (unfiltered)	$9.52 \times 10^{-2}$	$1.68 \times 10^{-2}$	$1.79 \times 10^{-3}$
ConforMixRMSD-Boltz (filtered)	$9.12 \times 10^{-2}$	$1.42 \times 10^{-2}$	$1.09 \times 10^{-3}$

Each entry in the table is the total fraction of bonds/angles/dihedrals marked as outliers, across all sampled frames and proteins.

Outlier analysis was performed using default settings of the `ramalyze` module of the `cctbx` toolkit, associated with Phenix ([4]), to identify Ramachandran outliers. We also evaluate backbone bond

length and bond angle outliers as deviations more than  $4\sigma$  from reference values. We assess every generated frame for outliers and compute the total fraction of bonds/angles/dihedrals that are marked as outliers, across all frames and proteins (Table S2). While the fraction of outliers is somewhat elevated in ConforMix sampling, the vast majority of bond lengths, bonds angles, and Ramachandran angles are within physical parameters. ConforMix-generated samples are generally usable as is for visual examination or motion statistics. If needed, most outliers are easily resolvable with a brief relaxation, as is standard in the AlphaFold workflow. For ConforMix, we also report results after our filtering, since we intentionally push some proteins beyond their normal range of flexibility in order to capture the full range of motion and reject unphysical samples afterwards.

## 7 Twisted diffusion sampling algorithm

The subroutine of twisted diffusion sampling is described below. The WEIGHT function accounts for the effects of the bias potential, and RESAMPLE outputs a new subset of the particles based on the weights. We use systematic resampling in our implementation. A more complete description of the algorithm is provided in [6].

---

### Algorithm S1 Twisted diffusion sampler for conformational generation

---

**Require:** Guidance potential  $y(x)$ , input system  $s$ , noise schedule  $\{\sigma_t\}$ , guidance scale schedule  $\{\epsilon_t\}$ , number of particles to generate  $N_{particles}$   
**Ensure:** Samples satisfying  $\text{MODEL}(x|y, s)$

- 1: **for**  $i = 1, \dots, N_{particles}$  **do**
- 2:    $x_i^T \in \mathbb{R}^{n_{atoms} \times 3} \sim \mathcal{N}(\vec{0}, \sigma_T \cdot I_3)$
- 3:    $w_i \leftarrow \text{WEIGHT}(x_i^T)$
- 4: **end for**
- 5: **for**  $t = T - 1, \dots, 0$  **do**
- 6:   **for**  $i = 1, \dots, N_{particles}$  **do**
- 7:      $\hat{x}_i^{t+1} \leftarrow \text{STRUCTUREDIFUSIONMODULE}(x_i^{t+1}, t + 1)$  // prediction of  $x_0$  at step  $t + 1$
- 8:      $\text{bias}_i^{t+1} = y(\hat{x}_i^{t+1})$
- 9:      $x_i^t \leftarrow x_i^{t+1} + \sigma_t(\hat{x}_i^{t+1} - x_i^{t+1}) + \epsilon_t(\nabla_{x_i^{t+1}} \log \text{bias}_i^{t+1})$  // backprop through model
- 10:     $w_i \leftarrow \text{WEIGHT}(x_i^t)$
- 11:   **end for**
- 12:    $x_{1:N}^t \sim \text{RESAMPLE}(x_{1:N}^t; w_{1:N})$
- 13: **end for**
- 14: **return**  $\{x_i^0\}_{i=1}^{N_{particles}}$

---

## 8 Free Energy Estimation with BioEmu details

The trp-cage-helix free energy estimation task described in the main text was performed using our implementation of ConforMix-BioEmu. The sampling procedure is similar to ConforMixRMSD (Algorithm 1), except that (1) we use the experimentally determined structure as a reference, and (2) we omit the secondary structure masking. This setup mimics common tasks in quantitative estimation of free energies, where one or more reference states can be described. For the particular free energy task described (estimation of  $\Delta G$  between 3Å and 7.5Å RMSD to the reference structure), we construct guidance potentials with an RMSD spacing of 0.5Å, between 2.0Å and 8.5Å from the reference. We collect 60 samples at each potential and perform free energy estimation with MBAR, as described in the text.

A tunable parameter in twisted diffusion sampling is the accuracy of the approximation  $\hat{x}_i^{t+1}$ . Because free energy estimation demands more accurate sampling than many qualitative tasks, we find it advantageous to substitute a multi-step denoised approximation (approximately 5 denoising steps) for the typical first- or second-order approximation used in the rest of our work. The performance cost of this approximation is on the order of a factor of 2. Because ConforMix free energy estimation aids in evaluating rare states, this penalty is still small compared to the alternative of running default sampling.

## 9 Supplemental Data and Figures

Table S3: AUC scores for coverage of experimentally-determined reference conformations by method and dataset

		Domain motion (n=38)	Membrane transporters (n=15)	Cryptic pockets (n=31)	Fold switching (n=15)
<b>Worst-matched reference conformation</b> (harder task)	Default Boltz sampling	0.37 ( $\pm 0.09$ )	0.16 ( $\pm 0.11$ )	0.23 ( $\pm 0.08$ )	0.13 ( $\pm 0.10$ )
	AFCluster-Boltz	0.45 ( $\pm 0.09$ )	0.26 ( $\pm 0.12$ )	0.42 ( $\pm 0.09$ )	0.22 ( $\pm 0.13$ )
	CF-random-Boltz	0.52 ( $\pm 0.09$ )	<b>0.28 (<math>\pm 0.13</math>)</b>	<b>0.43 (<math>\pm 0.10</math>)</b>	<b>0.24 (<math>\pm 0.12</math>)</b>
	AFsample2-Boltz	0.47 ( $\pm 0.09$ )	0.26 ( $\pm 0.14$ )	0.35 ( $\pm 0.10$ )	0.16 ( $\pm 0.13$ )
	<b>ConforMixRMSD-Boltz</b>	<b>0.61 (<math>\pm 0.08</math>)</b>	<b>0.28 (<math>\pm 0.13</math>)</b>	<b>0.43 (<math>\pm 0.09</math>)</b>	0.22 ( $\pm 0.11$ )
<b>Best-matched reference conformation</b> (easier task)	Default Boltz Sampling	<b>0.77 (<math>\pm 0.04</math>)</b>	0.49 ( $\pm 0.14$ )	0.80 ( $\pm 0.05$ )	0.54 ( $\pm 0.19$ )
	AFCluster-Boltz	0.73 ( $\pm 0.06$ )	0.49 ( $\pm 0.14$ )	<b>0.82 (<math>\pm 0.03</math>)</b>	0.50 ( $\pm 0.16$ )
	CF-random-Boltz	0.73 ( $\pm 0.07$ )	0.54 ( $\pm 0.11$ )	0.81 ( $\pm 0.06$ )	0.54 ( $\pm 0.17$ )
	AFsample2-Boltz	0.73 ( $\pm 0.06$ )	0.50 ( $\pm 0.14$ )	0.79 ( $\pm 0.06$ )	<b>0.57 (<math>\pm 0.20</math>)</b>
	<b>ConforMixRMSD-Boltz</b>	<b>0.77 (<math>\pm 0.04</math>)</b>	<b>0.56 (<math>\pm 0.12</math>)</b>	0.80 ( $\pm 0.04$ )	<b>0.57 (<math>\pm 0.16</math>)</b>

Coverage measures the fraction of proteins with samples matching a reference conformation within X% of the RMSD between reference structures. Displayed is the **area under the coverage curve**, AUC integrated from 0-100% of the reference-to-reference RMSD. Error bars are 95% confidence intervals over 1,000 bootstraps.

We evaluate coverage separately for best-matched and alternate (worst-matched) states. We use X% of the RMSD between reference structures, rather than RMSD directly, to normalize for the magnitude of each protein's conformational transition.

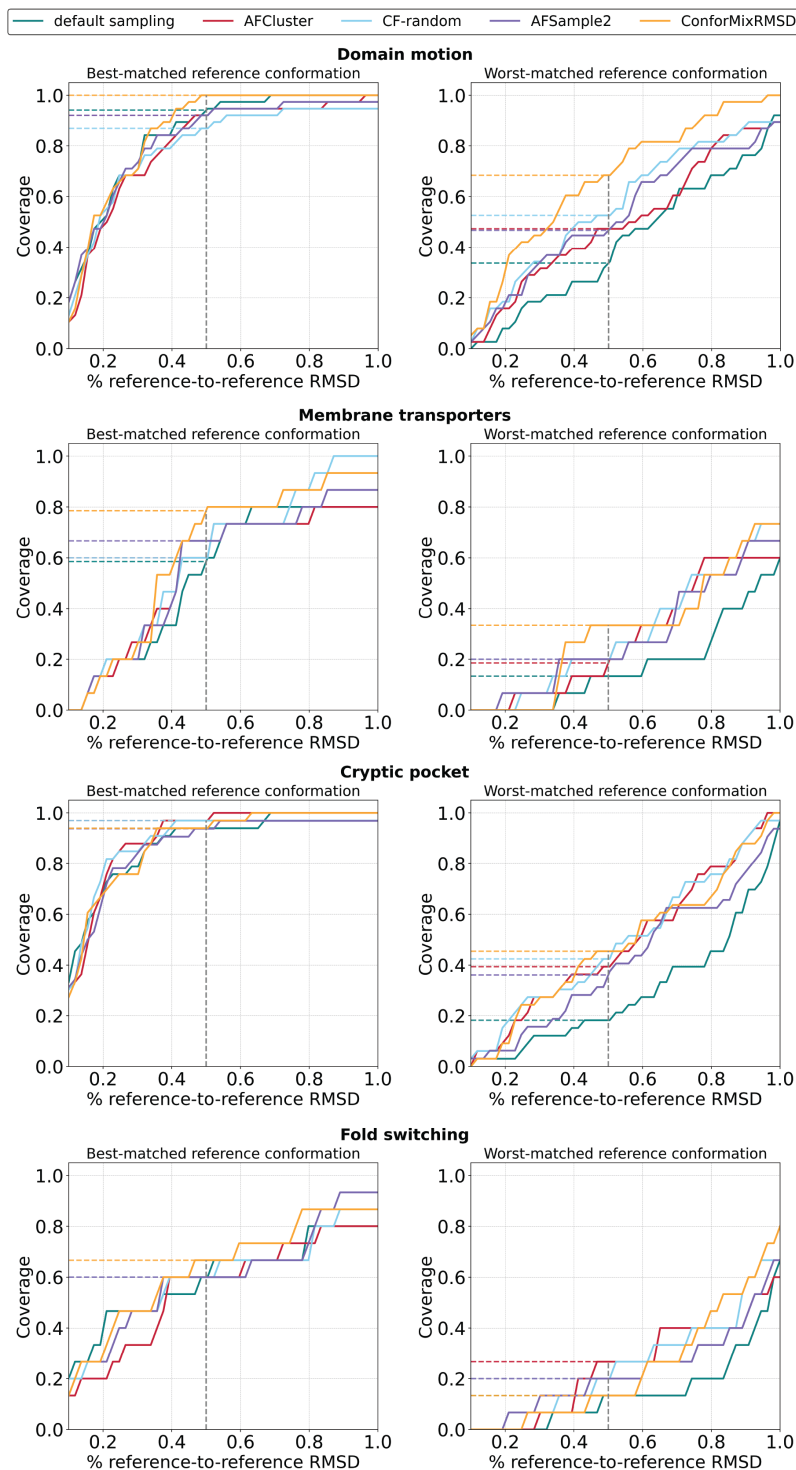


Figure S1: Coverage of experimentally observed states across methods and datasets. Coverage measure the fraction of proteins with samples matching a reference conformation within X% of the RMSD between reference structures. For each protein, we compute the RMSD of the closest sampled structure to each reference conformation. We then separate these into two groups: the best-matched reference (lower RMSD match) and the alternate reference (higher RMSD match). We aggregate performance across each dataset, evaluating coverage separately for best-matched and alternate (worst-matched) states. Coverage is computed at thresholds of X% of the RMSD between reference structures, normalizing for the magnitude of each protein's conformational transition.

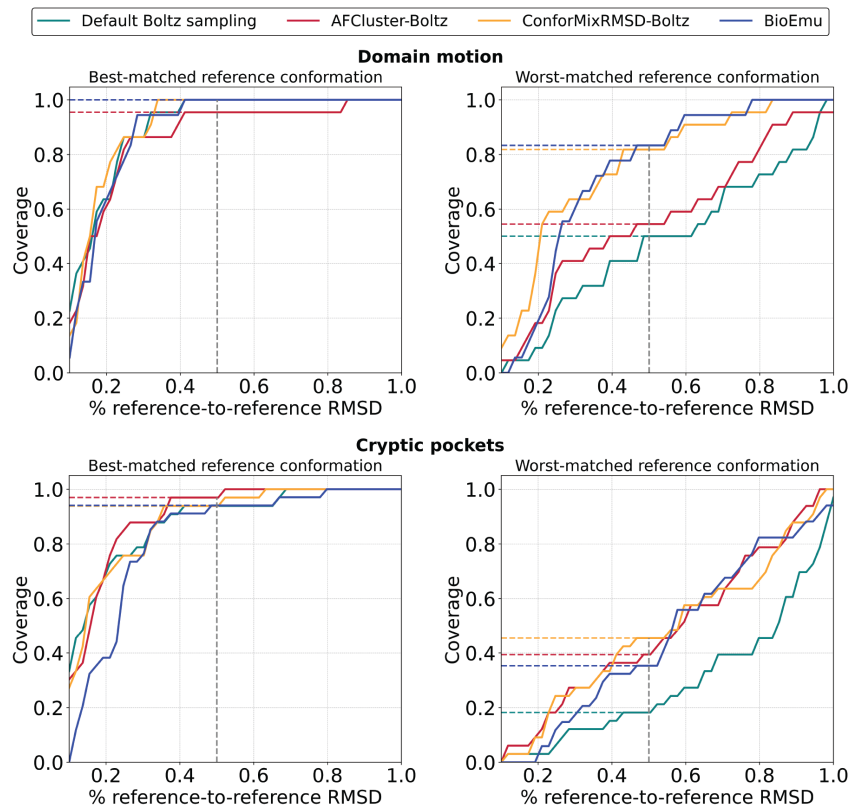


Figure S2: Comparison of coverage of experimentally observed states to BioEmu. Coverage measure the fraction of proteins with samples matching a reference conformation within X% of the RMSD between reference structures. For each protein, we compute the RMSD of the closest sampled structure to each reference conformation. We then separate these into two groups: the best-matched reference (lower RMSD match) and the alternate reference (higher RMSD match). We aggregate performance across each dataset, evaluating coverage separately for best-matched and alternate (worst-matched) states. Coverage is computed at thresholds of X% of the RMSD between reference structures, normalizing for the magnitude of each protein's conformational transition.

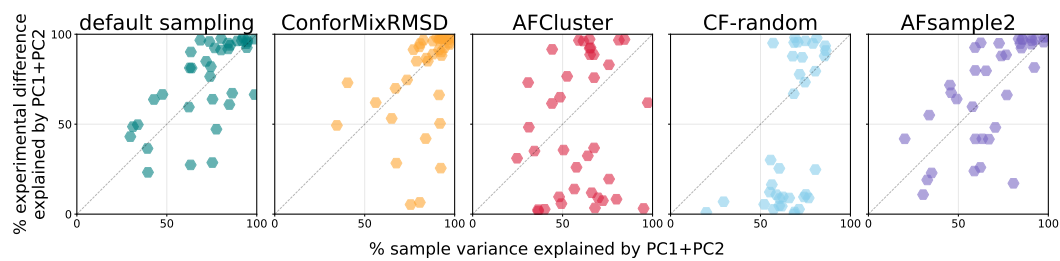


Figure S3: Principal component analysis of domain motion protein set, with extended benchmarks. Principal component analysis was performed pairwise  $C\alpha$  distances of structures sampled from each method. Each point describes results for one protein.



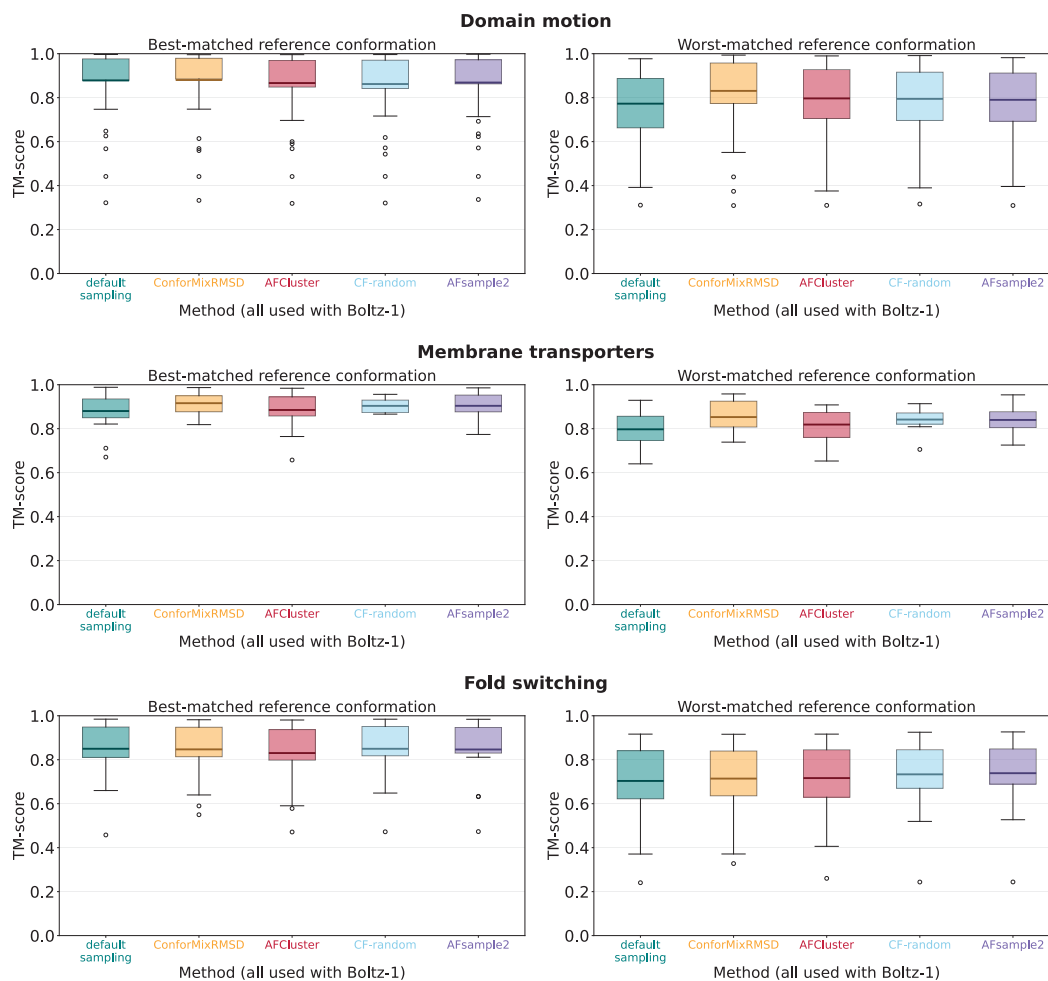


Figure S4: TM-score analysis across methods and datasets. TM-score is a length-independent structural similarity metric (0 – 1 scale), with higher values indicating closer structural matches.

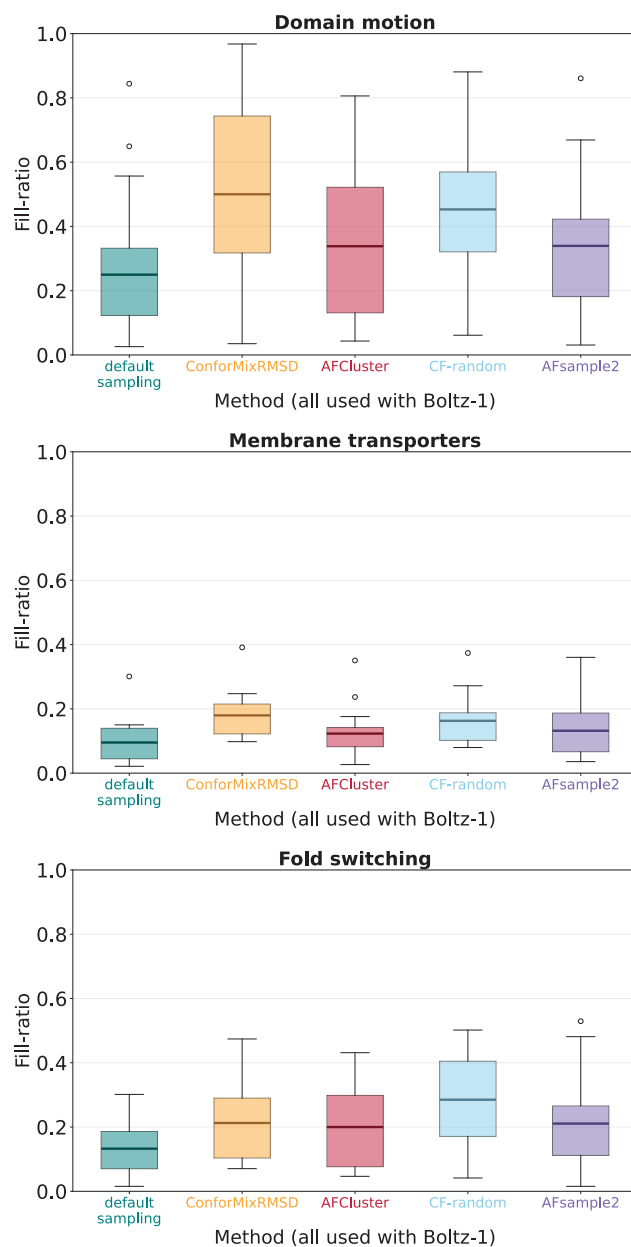


Figure S5: Fill ratio analysis across methods and datasets. The fill ratio, developed in [1], is a TM-score-based metric that quantifies the amount of structure diversity *between* reference states that is captured.

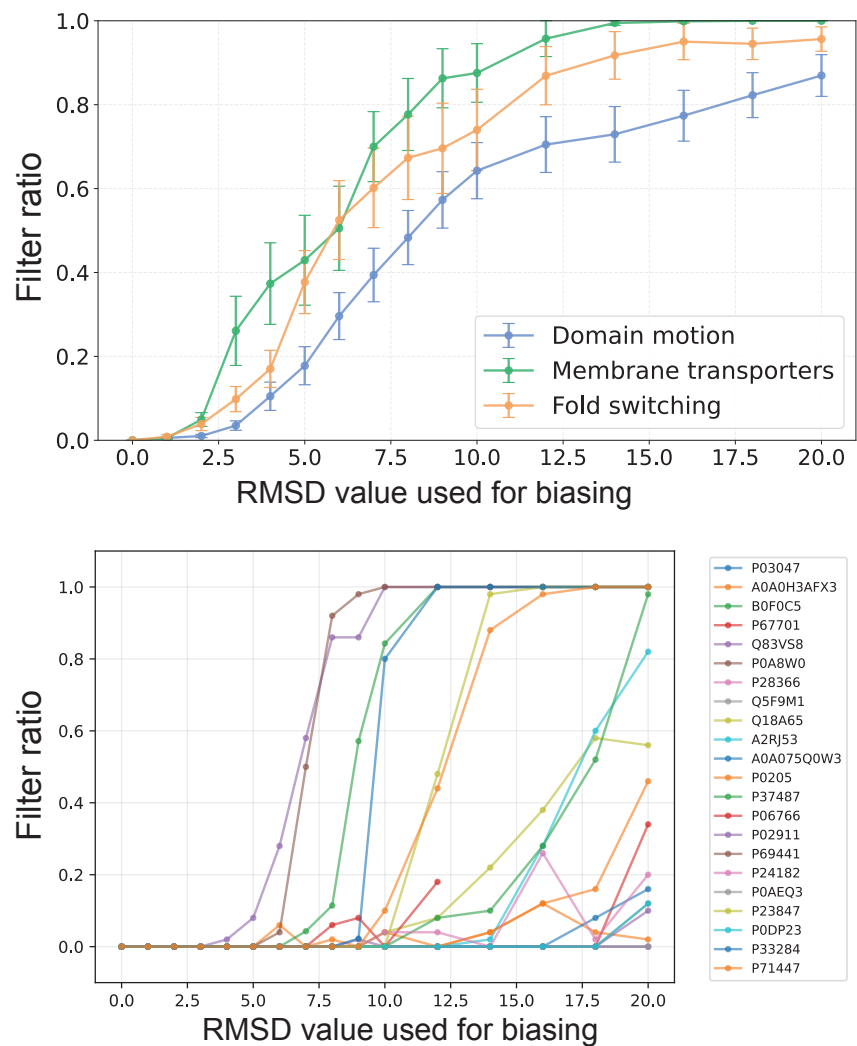


Figure S6: Filtering based on pLDDT scores with a 10-residue sliding rejects more samples as structure generation is biased further away from the default sample. The percent of ConforMixRMSD samples that are filtered out is plotted against the RMSD value against the default sample that used to condition sampling. Top: mean filter ratio across all proteins in each dataset. Error bars show the standard error of the mean. Bottom: filter ratio curves for a subset of domain motion proteins. Proteins tolerate varying amounts of biasing away from the default sample.



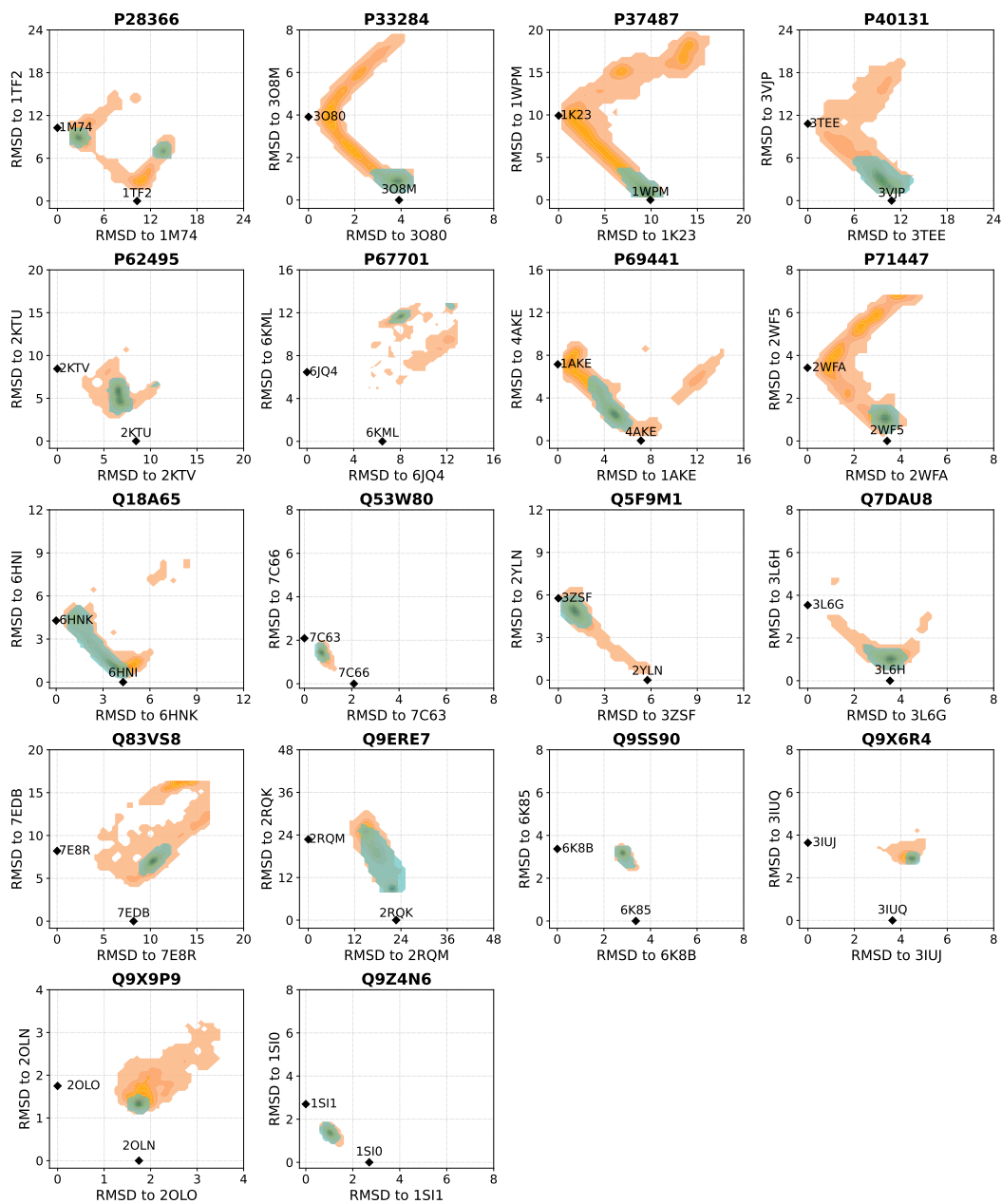


Figure S7 (continued): Density of sampling relative to reference experimentally-determined structures for **domain motion** dataset.

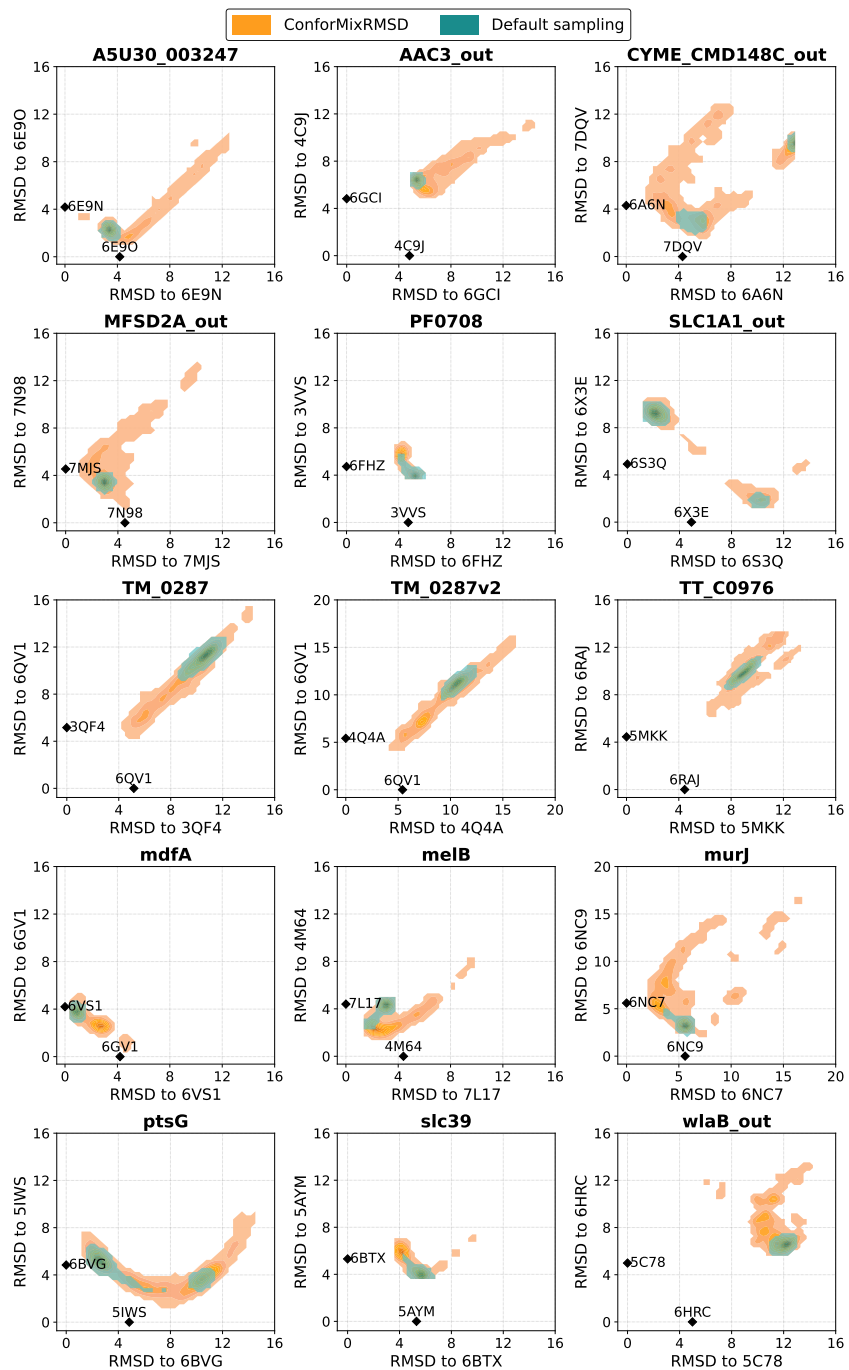


Figure S8: Density of sampling relative to reference experimentally-determined structures for **membrane transporters** dataset.

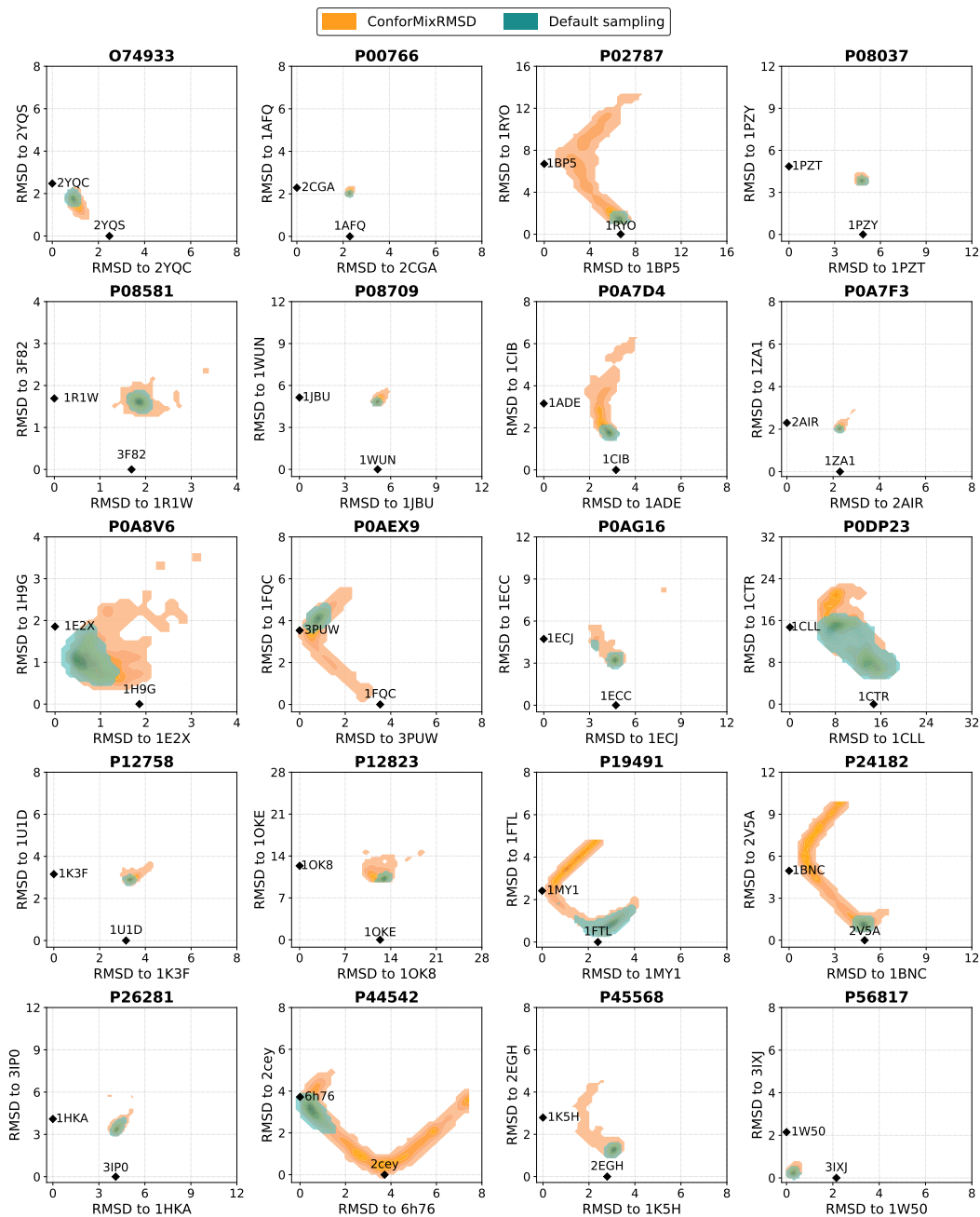


Figure S9: Density of sampling relative to reference experimentally-determined structures for **cryptic pocket** dataset.

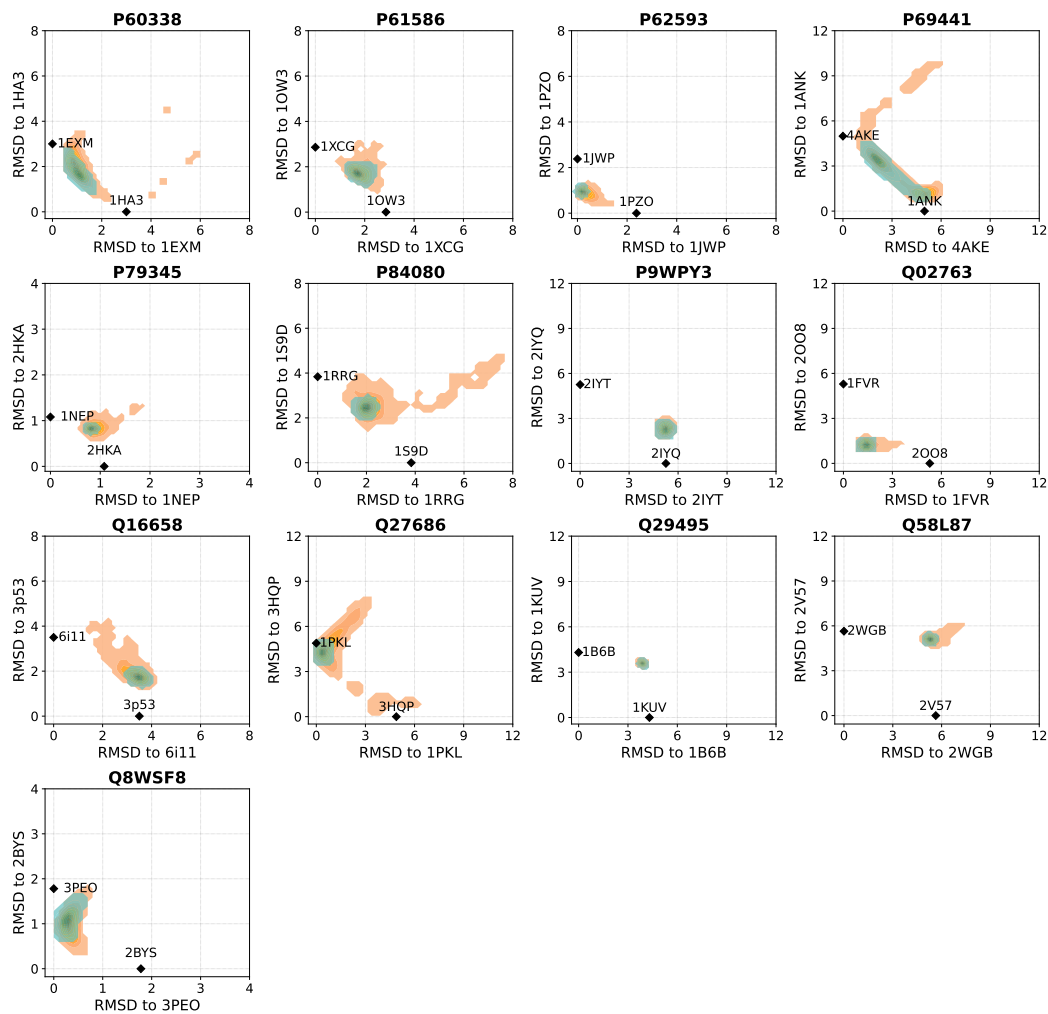


Figure S9 (continued): Density of sampling relative to reference experimentally-determined structures for **cryptic pocket** dataset.



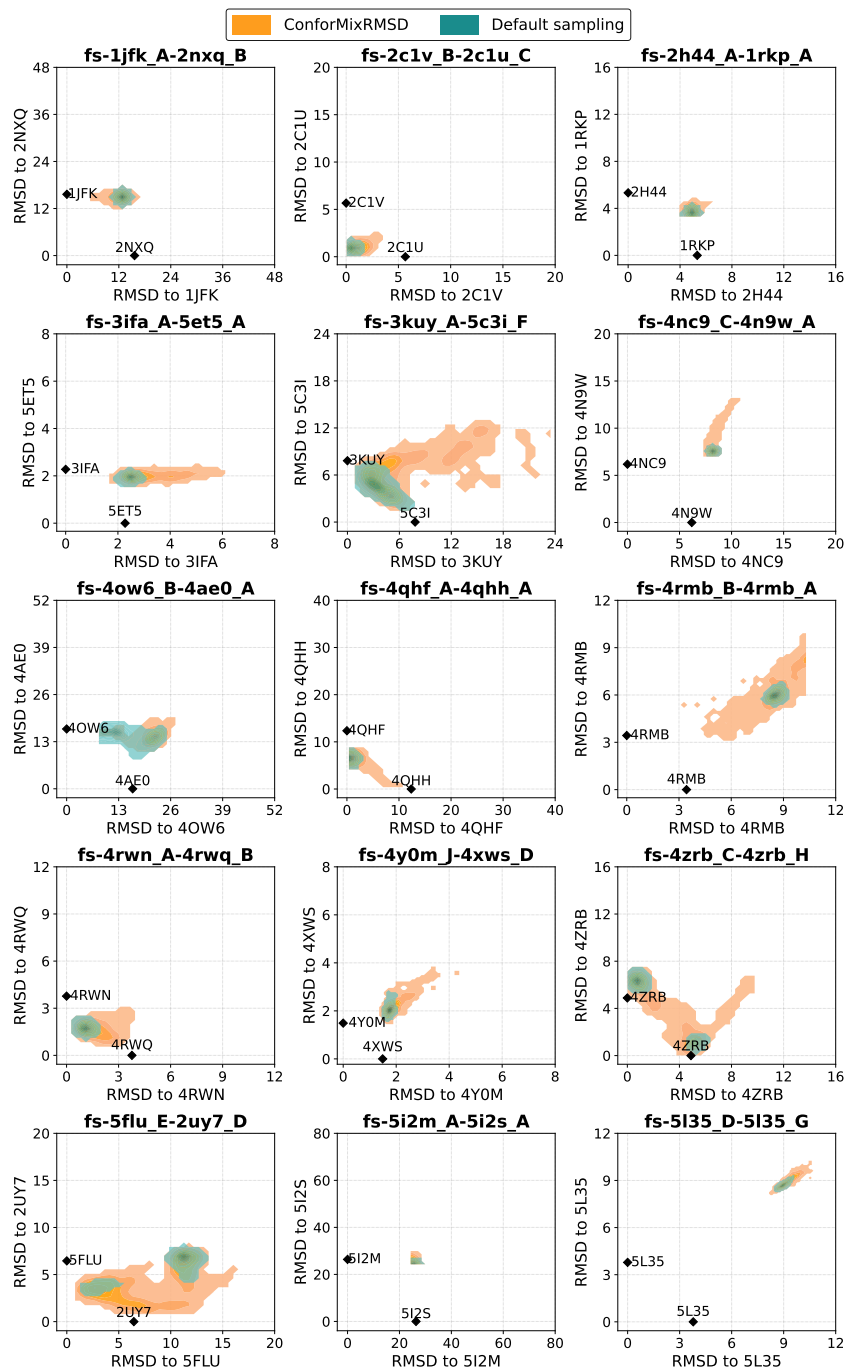


Figure S10: Density of sampling relative to reference experimentally-determined structures for **fold switching** dataset.

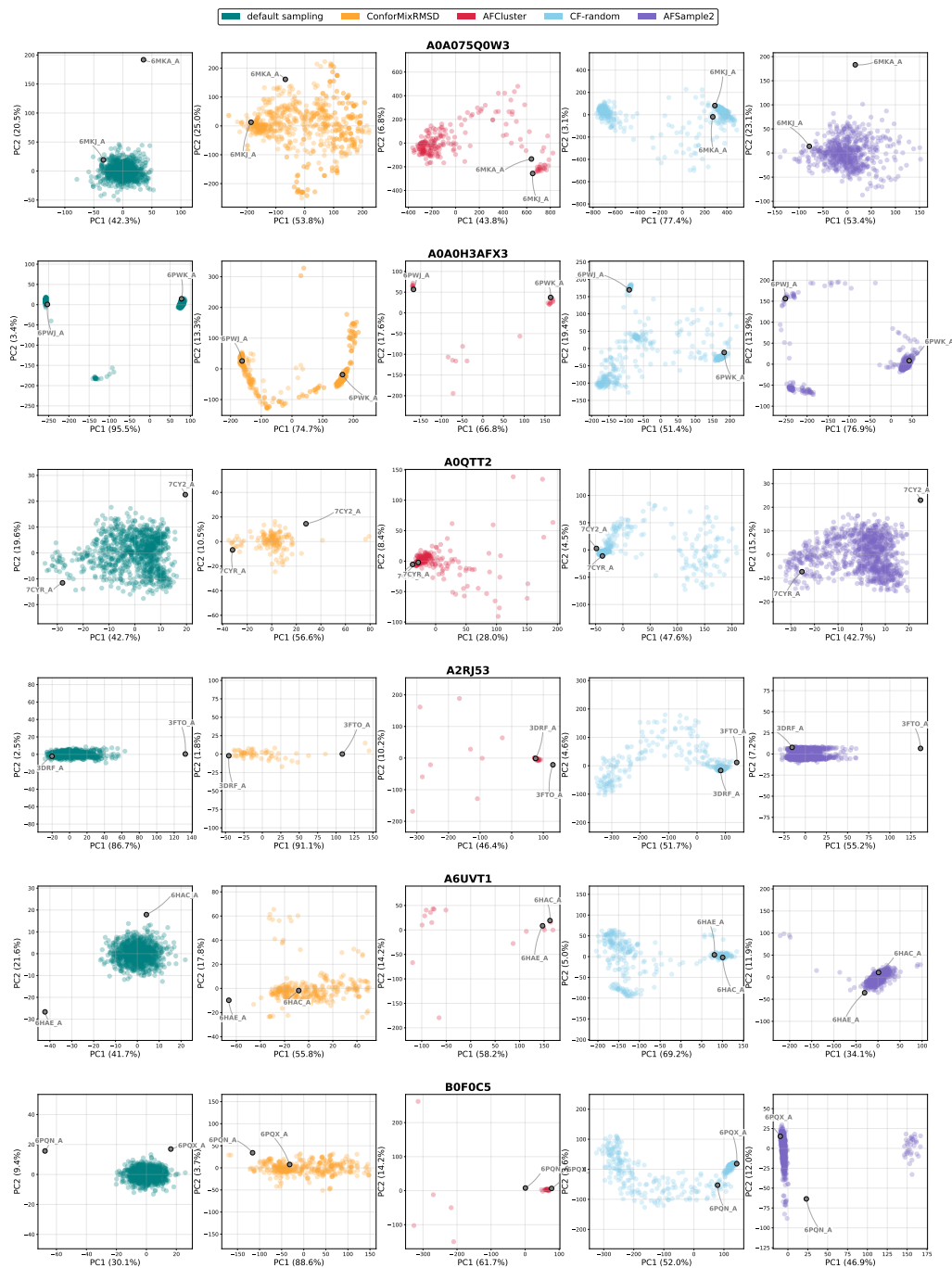


Figure S11: Principal component analysis of domain motion proteins.

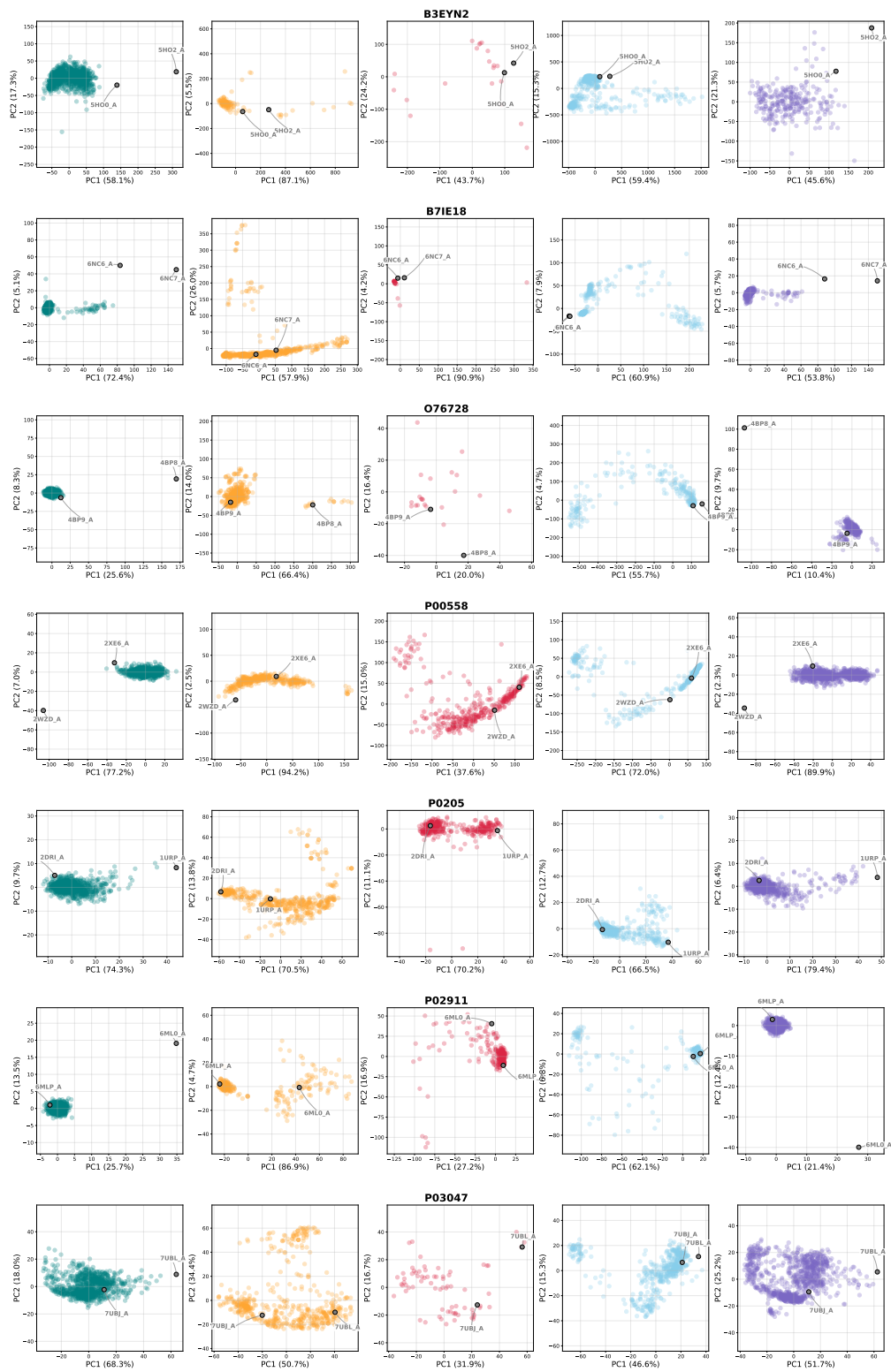


Figure S11 (continued): Principal component analysis of domain motion proteins.

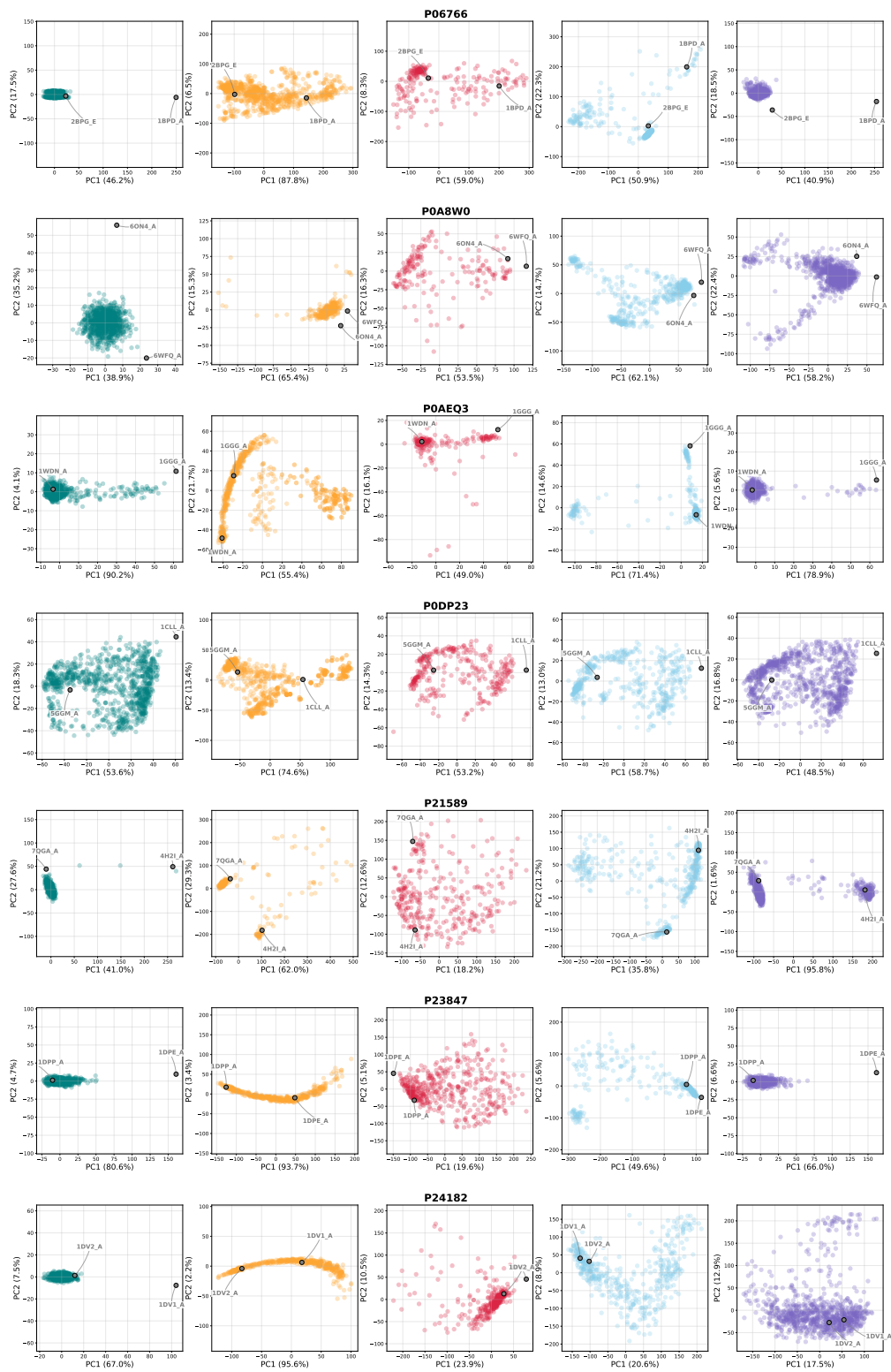


Figure S11 (continued): Principal component analysis of domain motion proteins.

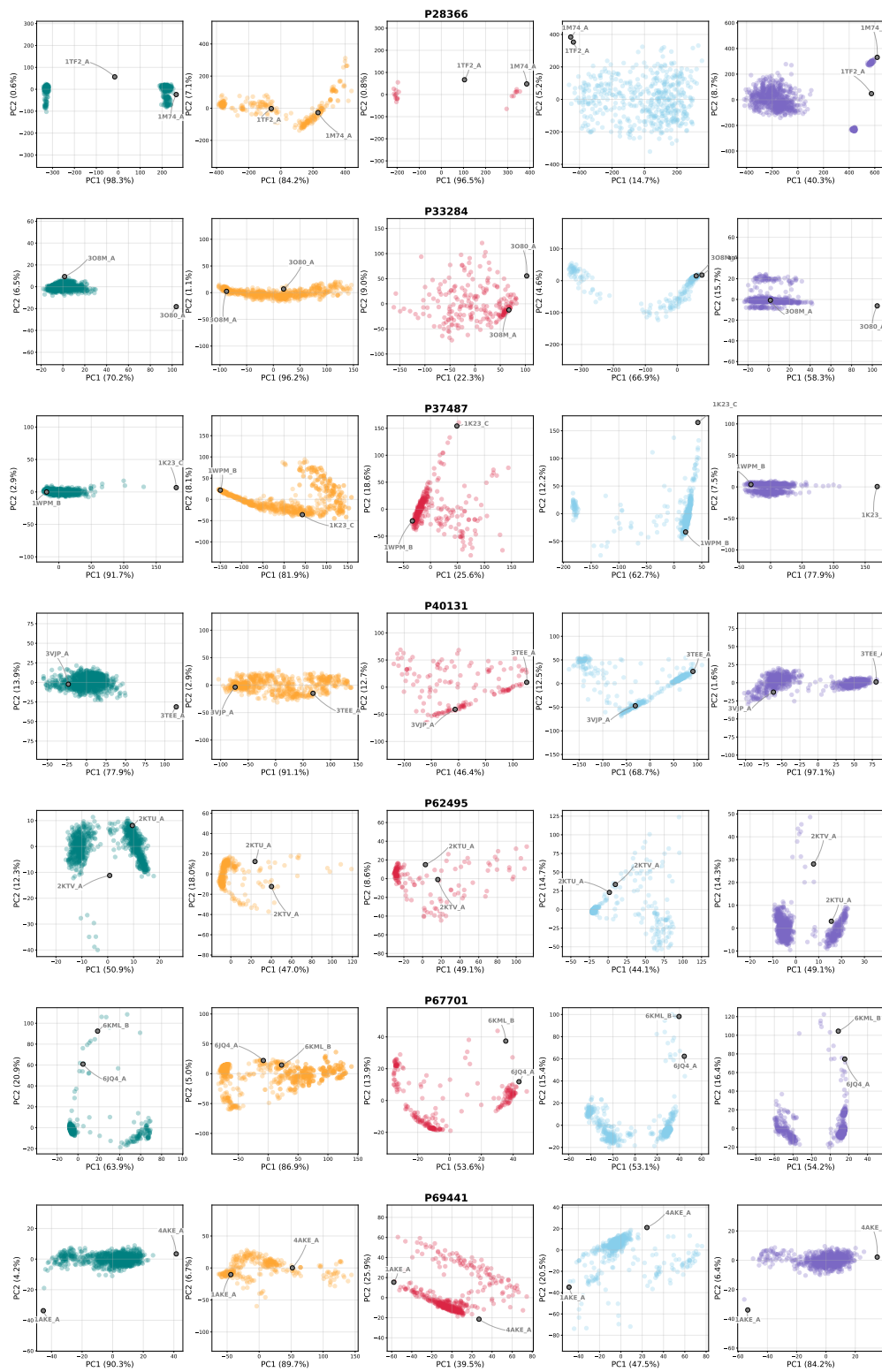


Figure S11 (continued): Principal component analysis of domain motion proteins.

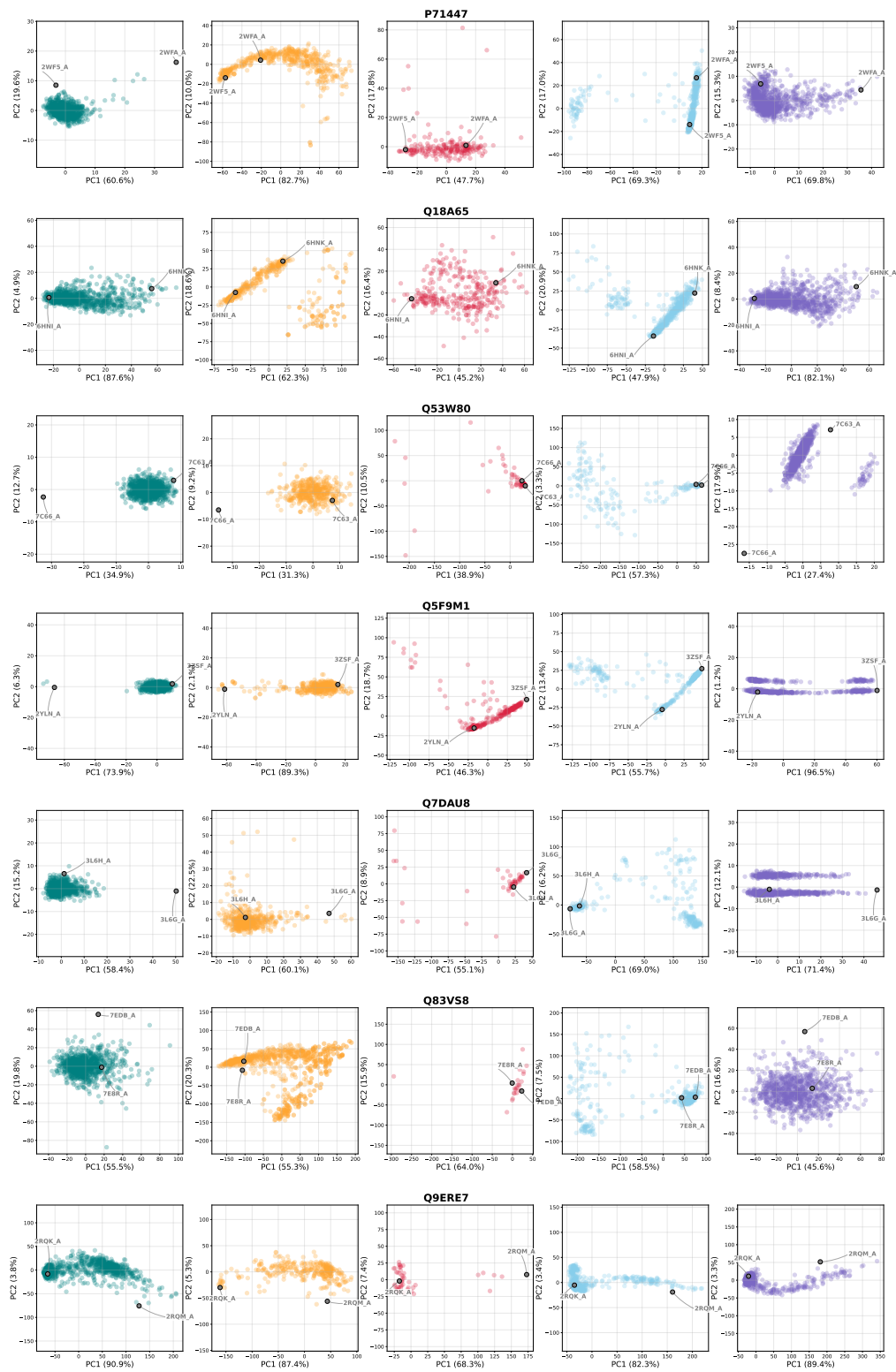


Figure S11 (continued): Principal component analysis of domain motion proteins.

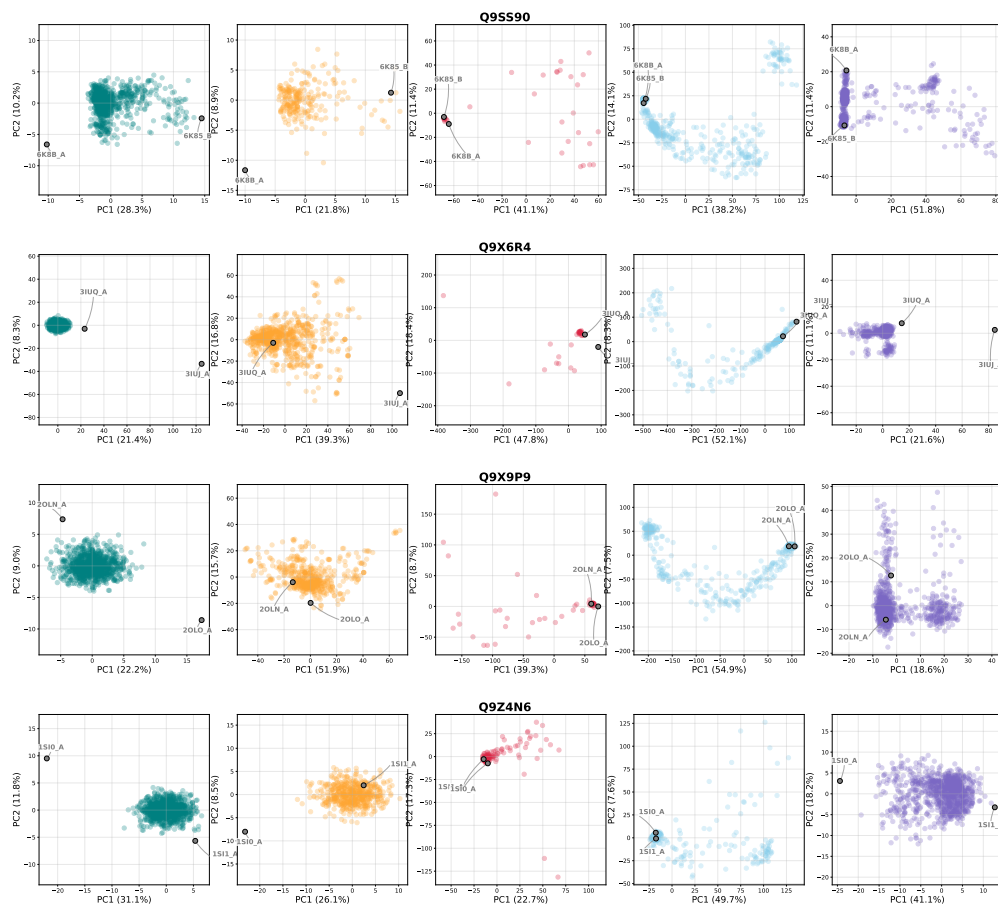


Figure S11 (continued): Principal component analysis of domain motion proteins.