

```
1 import Pkg, Revise; Pkg.activate(Base.current_project())
```



```
Activating project at `/DATA/cossio/SAM/2024/SamApp2024.jl`
```



```
1 import CairoMakie
```

```
1 import CSV
```

```
1 import Dates
```

```
1 import FASTX
```

```
1 import Infernal
```

```
1 import Makie
```

```
1 import Rfam
```

```
1 import SamApp2024
```

```
1 using BioSequences: LongRNA
```

```
1 using BioSequences: RNA_Gap
```

```
1 using SamApp2024: onehot
```

```
1 using DataFrames: DataFrame
```

```
1 using LinearAlgebra: eigen
```

```
1 using NaNStatistics: nanmean
```

```
1 using NaNStatistics: nansum
```

```
1 using RestrictedBoltzmannMachines: free_energy
```

```
1 using Statistics: cor
```

```
1 using Statistics: mean
```

```
1 shape_data = SamApp2024.load_shapemapper_data_500v2_20240315();
```

```
1 RF00162_hits = SamApp2024.rfam_RF00162_hits();
```

```
1 rbm_energies = free_energy(SamApp2024.rbm2022(), onehot(shape_data.aligned_sequences));
```

```
1 aptamer_natural_distances = SamApp2024.hamming(onehot(shape_data.aligned_sequences),  
onehot(RF00162_hits));
```

```
1 conds_sam = [1,2];
```

```
1 conds_mg = [4];
```

```
1 conds_30C = [6];
```

```
the_conds = ▶ [1, 2, 4, 6]
```

```
1 the_conds = vcat(conds_sam, conds_mg, conds_30C)
```

```
▶ ["SAMAPS2_1M7_0-1SAM_5Mg_T30C_rep0", "SAMAPS2_1M7_1SAM_5Mg_T30C_rep0", "SAMAPS2_1M7_noSAM_5Mg
```

```
1 shape_data.conditions[the_conds]
```

```
▶ (bps = [1, 2, 3, 4, 5, 6, 7, 8, 13, ... more ,108], nps = [9, 10, 11, 12, 18, 19, 20, 24, 32, ... more
```

```
1 bps, nps, pks = SamApp2024.RF00162_sites_paired()
```

```
ss_sites =
```

```
▶ (bps = [1, 2, 3, 4, 5, 6, 7, 8, 13, ... more ,108], nps = [9, 10, 11, 12, 18, 19, 20, 24, 32, ... more
```

```
1 ss_sites = SamApp2024.RF00162_sites_annotated_secondary_structure()
```

```
1 bps_reactivities = shape_data.shape_reactivities[bps, :, conds_sam];
```

```
1 nps_reactivities = shape_data.shape_reactivities[nps, :, conds_sam];
```

```
1 all_reactivities = shape_data.shape_reactivities[:, :, conds_sam];
```

```
1 shape_stats = SamApp2024.shape_basepair_log_odds_v4(  
2     shape_data = shape_data,  
3     paired_reactivities = bps_reactivities,  
4     unpaired_reactivities = nps_reactivities,  
5     all_reactivities = all_reactivities,  
6     only_hq_profile = true, p_thresh = 1e-2, nsamples=5000  
7 );
```

```
_sites =
```

```
▶ [25, 26, 27, 28, 77, 79, 34, 35, 36, 37, 73, 74, 75, 46, 47, 76, 100, 101, 102, 103, 104, 105]
```

```
1 _sites = SamApp2024.hallmark_sites_20230507
```

```
1 p4_pos = union(81:86, 92:97); # P4 helix positions
```

```
1 # length of P4 segment for probed sequences
```

```
2 aptamer_p4_length = length(p4_pos) .- [count(seq[p4_pos] .== RNA_Gap) for seq =  
shape_data.aligned_sequences];
```

```
1 x_mg = nansum(shape_stats.shape_log_odds[_sites, :, conds_mg]; dim=(1,3));
```

```
1 x_sam = nansum(shape_stats.shape_log_odds[_sites, :, conds_sam]; dim=(1,3));
```

```
_thresh = 1.6094379124341003
```

```
1 _thresh = log(5)
```

```
1 _responds_sam_yes = (x_mg .< -_thresh) .& (x_sam .> +_thresh);
```

```
1 _responds_sam_nop = (x_mg .> +_thresh) .| (x_sam .< -_thresh);
```

```
df =
```

	aligned_sequences	aptamer_origin	switch_yes	switch_no
1	108nt RNA Sequence: GUCUUUACUGAGCGGG-GAGGGACAGGCCUGUGAAC	"infernai"	false	true
2	108nt RNA Sequence: UUUUU----AGAGUAGCGGAGGGAUUGGCCCAUGAAG	"infernai"	false	true
3	108nt RNA Sequence: CCCUCAUCAAGAAUGGUGGAGGGACUGGCCCUAUGAAA	"infernai"	false	true
4	108nt RNA Sequence: UAGUCAUCCAGAGAGACAGAGGGACUGGACCAAUGAUG	"infernai"	false	true
5	108nt RNA Sequence: CACUUUAAAAGAGACGCGGAGGGACAGGCCCAAUGAAG	"infernai"	false	true
6	108nt RNA Sequence: UUUUUUAUCAAGAGCGGUGGAGGGAUUGGCCCUUGAAA	"infernai"	false	true
7	108nt RNA Sequence: CGGUCAUCCAGAGAAACCAAGGGAAAGGCCCAUGAAG	"infernai"	false	false
8	108nt RNA Sequence: AUCUUUUCGCGAGCAGUUGAGGGAUUGGCCCGUGAAG	"infernai"	false	true
9	108nt RNA Sequence: UACCUAUUCAGAGCGGCGGAGGGACUGGCUCUAUGAUA	"infernai"	false	false
10	108nt RNA Sequence: CUAUCAUCAAGAGGUGGAGAGGGACUGGCCCAUGAAC	"infernai"	false	true
⋮ more				
500	108nt RNA Sequence: AUUCGAUUACGACAGUCGGAGGGAACCAACCAUAGUAG	"Infrared"	false	false

```
1 df = DataFrame(  
2   aligned_sequences = shape_data.aligned_sequences,  
3   aptamer_origin = shape_data.aptamer_origin,  
4   switch_yes = _responds_sam_yes,  
5   switch_no = _responds_sam_nop  
6 )
```

```
1 rbm_seqs = findall(shape_data.aptamer_origin .== "rbm");
```

```
1 inf_seqs = findall(shape_data.aptamer_origin .== "infernai");
```

```
1 inf_untangled_seqs = findall(shape_data.aptamer_criteria .== "perm");
```

```
1 inf_uniform_seqs = findall(shape_data.aptamer_criteria .== "uniform");
```

```
1 rbm_p4_seqs = findall(shape_data.aptamer_criteria .== "p4");
```

```
1 # CM model from Rfam (this has the noisy floor!)  
2 Rfam_cm = Infernal.cmfetch(Rfam.cm(), "RF00162");
```

```
RF00162_seed_stk =
```

```
▶ (out = "/tmp/jl_LE30TkNeVe", stdout = "/tmp/jl_fkBqHqNyvs", stderr = "/tmp/jl_7sTuv2JsDy")
```

```
1 RF00162_seed_stk = Infernal.esl_afetch(Rfam.seed(), "RF00162")
```

```
1 RF00162_seed_match_cols = findall(≠('.'), SamApp2024.stockholm_ss(RF00162_seed_stk.out));
```

```
RF00162_seed_afa =
```

```
▶ (out = "/tmp/jl_FmbonQTOTq", stdout = "/tmp/jl_TULi5EBIrS", stderr = "/tmp/jl_Hk901UXCPW")
```

```
1 RF00162_seed_afa = Infernal.esl_reformat("AFA", RF00162_seed_stk.out;  
  informat="STOCKHOLM") # WARNING: this has inserts marked as '-'
```

```
RF00162_seed_records =
```

```
▶ [FASTX.FASTA.Record: , FASTX.FASTA.Record:  
  description: "AF027868.1/5245-5154" description: "AF269983.1/571-67
```

```
1 RF00162_seed_records = collect(FASTX.FASTA.Reader(open(RF00162_seed_afa.out)))
```

```
1 RF00162_seed_seqs_noinserts = LongRNA{4}.([FASTX.sequence(record)  
  [RF00162_seed_match_cols] for record in RF00162_seed_records]);
```

```
1 # trimmed (no inserts) aligned fasta  
2 RF00162_hits_afa = Infernal.cmalign(Rfam_cm.out, Rfam.fasta_file("RF00162");  
  matchonly=true, outformat="AFA");
```

```
1 # these are already aligned and without inserts  
2 RF00162_hits_sequences = LongRNA{4}.(FASTX.sequence.  
  (FASTX.FASTA.Reader(open(RF00162_hits_afa.out))));
```

```
1 # sites that have some non-zero fluctuations  
2 # We need to separate frozen sites below because otherwise cor and eigen give NaN,  
  infinities, and fail  
3 _variable_sites_flag = vec(all(0 <= mean(SamApp2024.onehot(RF00162_hits_sequences);  
  dims=3) <= 1; dims=1));
```

```
1 _variable_sites = findall(_variable_sites_flag);
```

```
1 RF00162_hits_var_sites_only = SamApp2024.onehot(RF00162_hits_sequences)[:,  
  _variable_sites, :];
```

```
1 RF00162_hits_cor = cor(reshape(RF00162_hits_var_sites_only, :,  
  size(RF00162_hits_var_sites_only, 3)); dims=2);
```

```
1 RF00162_hits_eig = eigen(RF00162_hits_cor);
```

```

0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

[:, :, 3] =
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

;;; ...

```

```

[:, :, 438] =
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

[:, :, 439] =
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

[:, :, 440] =
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ... 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

```

```

1 # remap the variable sites eigenvectors back to the original consensus sequence numbering
2 RF00162_hits_eigvec = zeros(5, 108, size(RF00162_hits_eig.vectors, 1))

```

```

1 for n in 1:size(RF00162_hits_eig.vectors, 1)
2     vec(view(RF00162_hits_eigvec, :, _variable_sites, n)) .= RF00162_hits_eig.vectors[:,
n]
3 end

```

```

1 shape_sequences_onehot = SamApp2024.onehot(LongRNA{4}.(shape_data.aligned_sequences));

```

```

1 __proj_500 = reshape(shape_sequences_onehot, 5*108, :)' * reshape(RF00162_hits_eigvec,
5*108, :);

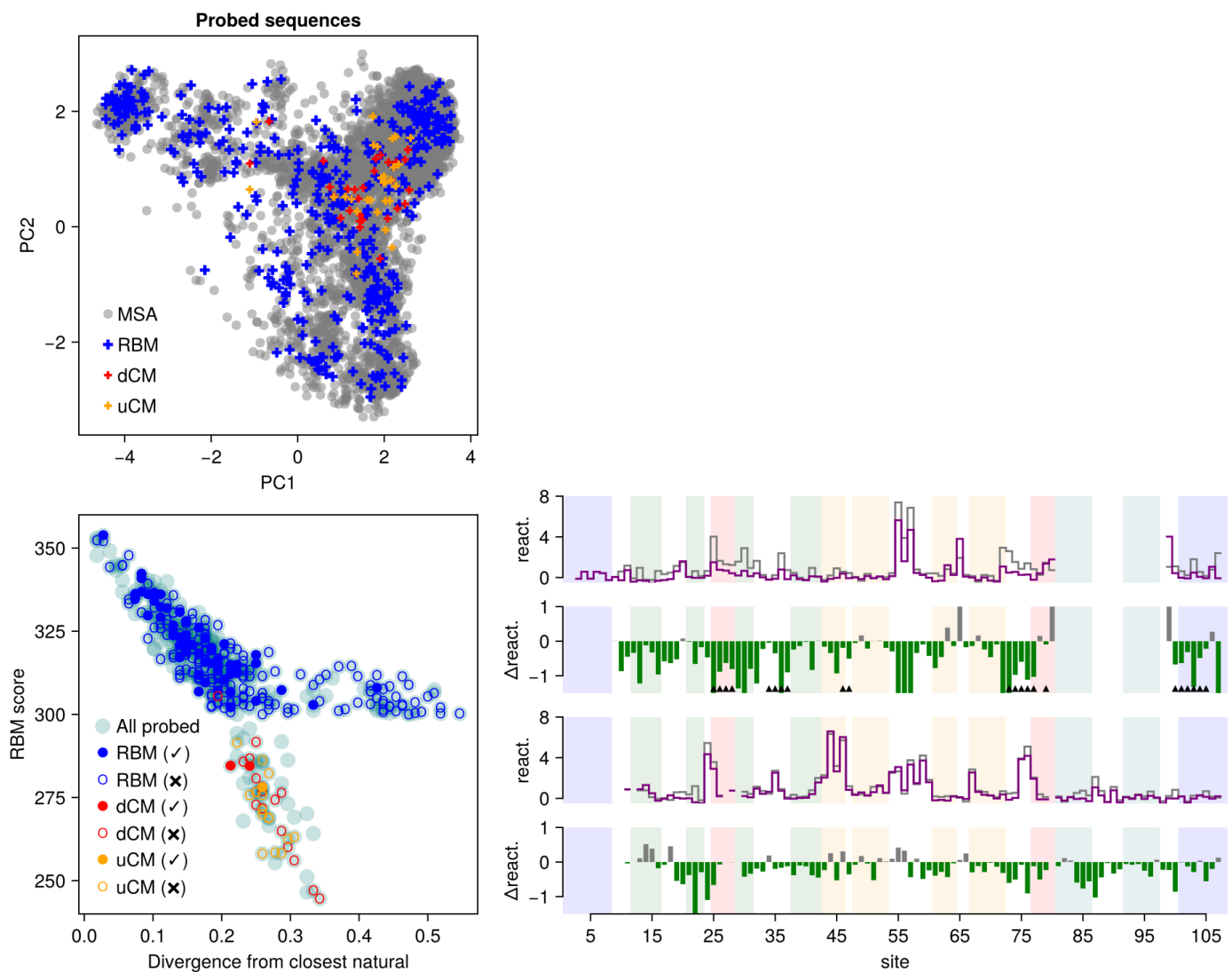
```

```

1 __proj_hits = reshape(SamApp2024.onehot(RF00162_hits_sequences), 5*108, :)' *
reshape(RF00162_hits_eigvec, 5*108, :);

```

```
1 # structural motifs
2 struct_bands = [
3     (; x0=0.5, xf=8.5, color="blue", alpha=0.1), # P1
4     (; x0=100.5, xf=108.5, color="blue", alpha=0.1), # P1
5     (; x0=11.5, xf=16.5, color="green", alpha=0.1), # P2
6     (; x0=20.5, xf=23.5, color="green", alpha=0.1), # P2
7     (; x0=28.5, xf=31.5, color="green", alpha=0.1), # P2
8     (; x0=37.5, xf=42.5, color="green", alpha=0.1), # P2
9     (; x0=42.5, xf=46.5, color="orange", alpha=0.1), # P3
10    (; x0=47.5, xf=53.5, color="orange", alpha=0.1), # P3
11    (; x0=60.5, xf=64.5, color="orange", alpha=0.1), # P3
12    (; x0=66.5, xf=72.5, color="orange", alpha=0.1), # P3
13    (; x0=80.5, xf=86.5, color="teal", alpha=0.1), # P4
14    (; x0=91.5, xf=97.5, color="teal", alpha=0.1), # P4
15    (; x0=24.5, xf=28.5, color="red", alpha=0.1), # Pk
16    (; x0=76.5, xf=80.5, color="red", alpha=0.1), # Pk
17 ];
```



```

1 let fig = Makie.Figure()
2 ax = Makie.Axis(fig[1,1][1,1], xlabel="PC1", ylabel="PC2", width=300, height=300,
3 xgridvisible=false, ygridvisible=false, title="Probed sequences")
4 Makie.scatter!(ax, __proj_hits[:, end], __proj_hits[:, end - 1], markersize=10,
5 color=:gray, 0.5), label="MSA")
6 # Makie.scatter!(ax,
7 #   __proj_500[(aptamer_p4_length .< 1) .& _responds_sam_yes, end],
8 #   __proj_500[(aptamer_p4_length .< 1) .& _responds_sam_yes, end - 1],
9 #   markersize=15, color=:orange, marker='●'
10 # )
11 Makie.scatter!(ax,
12   __proj_500[shape_data.aptamer_origin .== "rbm", end],
13   __proj_500[shape_data.aptamer_origin .== "rbm", end - 1],
14   markersize=10, color=:blue, label="RBM", marker=:cross
15 )
16 Makie.scatter!(ax,
17   __proj_500[inf_uniform_seqs, end],
18   __proj_500[inf_uniform_seqs, end - 1],
19   markersize=8, color=:red, label="dCM", marker=:cross
20 )
21 Makie.scatter!(ax,
22   __proj_500[inf_untangled_seqs, end],
23   __proj_500[inf_untangled_seqs, end - 1],
24   markersize=8, color=:orange, label="uCM", marker=:cross
25 )
26 Makie.axislegend(ax, position=:lb, framevisible=false, patchlabelgap=-3)

```

```

27 ax = Makie.Axis(fig[1,1][2,1], width=300, height=300, xlabel="Divergence from
closest natural", ylabel="RBM score", xticks=0:0.1:0.6, yticks=200:25:350,
xgridvisible=false, ygridvisible=false)
28 Makie.scatter!(ax, vec(minimum(aptamer_natural_distances; dims=2)) / 108, -
rbm_energies, markersize=15, color=(teal, 0.2), marker='●', label="All probed")
Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[rbm_seqs ∩
29 findall(_responds_sam_yes), :]; dims=2)) / 108, -rbm_energies[rbm_seqs ∩
findall(_responds_sam_yes)], markersize=10, color=:blue, marker='●', label="RBM (✓)")
Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[rbm_seqs ∩
30 findall(_responds_sam_nop), :]; dims=2)) / 108, -rbm_energies[rbm_seqs ∩
findall(_responds_sam_nop)], markersize=10, color=:blue, marker='○', label="RBM (✗)")
# Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[inf_seqs ∩
31 findall(_responds_sam_yes), :]; dims=2)), -rbm_energies[inf_seqs ∩
findall(_responds_sam_yes)], markersize=10, color=:red, marker='●', label="CM (✓)")
# Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[inf_seqs ∩
32 findall(_responds_sam_nop), :]; dims=2)), -rbm_energies[inf_seqs ∩
findall(_responds_sam_nop)], markersize=10, color=:red, marker='○', label="CM (✗)")
Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[inf_uniform_seqs ∩
33 findall(_responds_sam_yes), :]; dims=2)) / 108, -rbm_energies[inf_uniform_seqs ∩
findall(_responds_sam_yes)], markersize=10, color=:red, marker='●', label="dCM (✓)")
Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[inf_uniform_seqs ∩
34 findall(_responds_sam_nop), :]; dims=2)) / 108, -rbm_energies[inf_uniform_seqs ∩
findall(_responds_sam_nop)], markersize=10, color=:red, marker='○', label="dCM (✗)")
Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[inf_untangled_seqs ∩
35 findall(_responds_sam_yes), :]; dims=2)) / 108, -rbm_energies[inf_untangled_seqs ∩
findall(_responds_sam_yes)], markersize=10, color=:orange, marker='●', label="uCM
(✓)")
Makie.scatter!(ax, vec(minimum(aptamer_natural_distances[inf_untangled_seqs ∩
findall(_responds_sam_nop), :]; dims=2)) / 108, -rbm_energies[inf_untangled_seqs ∩
36 findall(_responds_sam_nop)], markersize=10, color=:orange, marker='○', label="uCM
(✗)")
37 Makie.ylims!(ax, 240, 360)
Makie.axislegend(ax, position=(-0.02, -0.01), framevisible=false, nbanks=1,
38 colgap=1, rowgap=0.1, patchlabelgap=0)
39
40
41 _dummy_axis = Makie.Axis(fig[1,2][0,1])
42 Makie.hidespines!(_dummy_axis, :t, :b, :l, :r)
43 Makie.hidexdecorations!(_dummy_axis)
44 Makie.hideydecorations!(_dummy_axis)
45
46 _width = 500
47 _height = 65
48
49 # example of no P4
50 n_ex = 116
51
52 _R_mg = shape_data.shape_reactivities[:, n_ex, only(conds_mg)]
53 _R_sam = shape_data.shape_reactivities[:, n_ex, conds_sam[2]]
54
55 ax_react_1 = Makie.Axis(fig[1,2][1,1]; valign=:bottom, width=_width, height=_height,
xticks=5:10:108, ylabel="react.", xgridvisible=false, ygridvisible=false,
56 yticks=0:4:8, xtrimspine=true, ytrimspine=true)
57 for (x0, xf, color, alpha) = struct_bands
58 Makie.vspan!(ax_react_1, x0, xf; color=(color, alpha))
59 end
Makie.stairs!(ax_react_1, 1:108, _R_mg, color=:gray, step=:center, label="no SAM")
Makie.stairs!(ax_react_1, 1:108, _R_sam, color=:purple, step=:center, label="with
60 SAM")
61 Makie.hidespines!(ax_react_1, :t, :r, :b)
62

```



```

63 Makie.hidexdecorations!(ax_react_1)
64 #Makie.axislegend(ax_react_1, position=(0.0, -13), framevisible=false)

ax_diff_1 = Makie.Axis(fig[1,2][2,1]; valign=:bottom, width=_width, height=_height,
65 xticks=5:10:108, xlabel="site", ylabel="Δreact.", xgridvisible=false,
66 ygridvisible=false, yticks=-1:1, xtrimspine=true, ytrimspine=true)
67 for (x0, xf, color, alpha) = struct_bands
68     Makie.vspan!(ax_diff_1, x0, xf; color=(color, alpha))
69 end
69 Makie.barplot!(ax_diff_1, 1:108, _R_sam - _R_mg, color=ifelse.(_R_sam - _R_mg .< 0,
:gray))
70 Makie.scatter!(ax_diff_1, _sites, -1.4one.(_sites), markersize=7, color=:black,
71 marker=:utriangle)
72 Makie.xlims!(ax_diff_1, 0, 109)
73 Makie.hidespines!(ax_diff_1, :r, :b, :t)
74 Makie.hidexdecorations!(ax_diff_1)
75 #Makie.scatter!(ax_diff_1, _sites, -0.2one.(_sites), color=:blue, markersize=5)
76
77 # example distant
78 n_ex = 284

79
80 _R_mg = shape_data.shape_reactivities[:, n_ex, only(conds_mg)]
81 _R_sam = shape_data.shape_reactivities[:, n_ex, conds_sam[2]]

ax_react_2 = Makie.Axis(fig[1,2][3,1]; valign=:bottom, width=_width, height=_height,
82 xticks=5:10:108, yticks=0:4:8, ylabel="react.", xgridvisible=false,
83 ygridvisible=false, xtrimspine=true, ytrimspine=true)
84 for (x0, xf, color, alpha) = struct_bands
85     Makie.vspan!(ax_react_2, x0, xf; color=(color, alpha))
86 end
87 Makie.stairs!(ax_react_2, 1:108, _R_mg, color=:gray, step=:center, label="no SAM")
88 Makie.stairs!(ax_react_2, 1:108, _R_sam, color=:purple, step=:center, label="with
SAM")
89 Makie.hidespines!(ax_react_2, :t, :r, :b)
90 Makie.hidexdecorations!(ax_react_2)

ax_diff_2 = Makie.Axis(fig[1,2][4,1]; valign=:bottom, width=_width, height=_height,
91 xticks=5:10:108, yticks=-1:1, xlabel="site", ylabel="Δreact.", xgridvisible=false,
92 ygridvisible=false, xtrimspine=true, ytrimspine=true)
93 for (x0, xf, color, alpha) = struct_bands
94     Makie.vspan!(ax_diff_2, x0, xf; color=(color, alpha))
95 end
96 Makie.barplot!(ax_diff_2, 1:108, _R_sam - _R_mg, color=ifelse.(_R_sam - _R_mg .< 0,
:gray))
97 Makie.hidespines!(ax_diff_2, :r, :t)
98 #Makie.scatter!(ax_diff_2, _sites, -0.2one.(_sites), color=:blue, markersize=5)
99

100 Makie.linkxaxes!(ax_react_1, ax_diff_1)
101 Makie.linkxaxes!(ax_react_2, ax_diff_2)
102 Makie.linkyaxes!(ax_react_1, ax_react_2)
103 Makie.linkyaxes!(ax_diff_1, ax_diff_2)
104
105 Makie.ylims!(ax_diff_1, -1.5, 1)
106 Makie.ylims!(ax_diff_2, -1.5, 1)
107
108 Makie.ylims!(ax_react_1, -0.5, 8)
109 Makie.ylims!(ax_react_2, -0.5, 8)
110
111 Makie.xlims!(ax_react_1, 0.5, 108.5)
112 Makie.xlims!(ax_react_2, 0.5, 108.5)

```

```
113 Makie.xlims!(ax_diff_1, 0.5, 108.5)
114 Makie.xlims!(ax_diff_2, 0.5, 108.5)
115
116 Makie.resize_to_layout!(fig)
```