

```
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 FASTX
```

```
1 import HDF5
```

```
1 import Infernal
```

```
1 import KernelDensity
```

```
1 import Makie
```

```
1 import RestrictedBoltzmannMachines as RBMs
```

```
1 import Rfam
```

```
1 import SamApp2024
```

```
1 import StatsBase
```

```
1 import ViennaRNA
```

```
1 import ViennaRNA_jll
```

```
1 using BioSequences: LongRNA
```

```
1 using DataFrames: DataFrame
```

```
1 using Distributions: Gamma
```

```
1 using Distributions: logpdf
```

```
1 using Distributions: pdf
```

```
1 using Distributions: Poisson
```

```
1 using LinearAlgebra: Diagonal
```

```
1 using LinearAlgebra: eigen
```

```
1 using Makie: @L_str
```

```
1 using NaNStatistics: nanmean, nansum
```

```
1 using Random: bitrand
```

```
1 using RestrictedBoltzmannMachines: free_energy
```

```
1 using Statistics: cor
```

```
1 using Statistics: mean
```

```
1 using Statistics: middle
```

```
1 using StatsBase: countmap
```

```
1 using Unitful: ustrip
```

```
1 @show Rfam.get_rfam_directory();
```

```
> Rfam.get_rfam_directory() = "/DATA/rossio/data/Rfam" ⓘ
```

```
1 @show Rfam.get_rfam_version();
```

```
> Rfam.get_rfam_version() = "14.7" ⓘ
```

```
1 # load SHAPE data
2 shape_data_045 = SamApp2024.load_shapemapper_data_pierre_demux_20230920(; demux=true);
```

```
1 # split rep0 from rep4+5
2 shape_data_rep0 = SamApp2024.select_conditions_20231002(shape_data_045,
  filter(endswith("_rep0"), shape_data_045.conditions));
```

```
1 # split rep0 from rep4+5
2 shape_data_rep45 = SamApp2024.select_conditions_20231002(shape_data_045,
  filter(endswith("_rep45"), shape_data_045.conditions));
```

```
1 conds_sam_rep0 = identity.(indexin(["SAMAP_1M7_0-1SAM_5Mg_T30C_rep0", "SAMAP_1M7_0-
  5SAM_5Mg_T30C_rep0", "SAMAP_1M7_1SAM_5Mg_T30C_rep0"], shape_data_rep0.conditions));
```

```
1 conds_mg_rep0 = identity.(indexin(["SAMAP_1M7_noSAM_5Mg_T30C_rep0"],
  shape_data_rep0.conditions));
```

```
1 conds_30C_rep0 = identity.(indexin(["SAMAP_1M7_noSAM_noMg_T30C_rep0"],
  shape_data_rep0.conditions));
```

```
1 conds_sam_rep45 = identity.(indexin(["SAMAP_1M7_0-1SAM_5Mg_T30C_rep45",
  "SAMAP_1M7_1SAM_5Mg_T30C_rep45"], shape_data_rep45.conditions));
```

```
1 conds_mg_rep45 = identity.(indexin(["SAMAP_1M7_noSAM_5Mg_T30C_rep45"],
  shape_data_rep45.conditions));
```

```
1 conds_30C_rep45 = identity.(indexin(["SAMAP_1M7_noSAM_noMg_T30C_rep45"],
  shape_data_rep45.conditions));
```

```
1 @show conds_sam_rep0 conds_mg_rep0 conds_30C_rep0;
```

```
> conds_sam_rep0 = [1, 2, 3]
   conds_mg_rep0 = [4]
   conds_30C_rep0 = [5]
```

```
1 @show conds_sam_rep45 conds_mg_rep45 conds_30C_rep45;
```

```
> conds_sam_rep45 = [1, 2]
   conds_mg_rep45 = [4]
   conds_30C_rep45 = [6]
```

```
► (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()
```

```
rbm_seqs =
```

```
► [207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 22
```

```
1 rbm_seqs = findall(shape_data_045.aptamer_origin .== "RF00162_syn_rbm")
```

```
inf_seqs = ► [258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 291]
```

```
1 inf_seqs = findall(shape_data_045.aptamer_origin .== "RF00162_syn_inf")
```

```
full_seqs =
```

```
► [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ... more ,46, 47, 48, 49, 50,
```

```
1 full_seqs = findall(shape_data_045.aptamer_origin .== "RF00162_full30")
```

```
seed_seqs =
```

```
► [56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, ... more ,197, 198,
```

```
1 seed_seqs = findall(shape_data_045.aptamer_origin .== "RF00162_seed70")
```

```
1 nat_seqs = full_seqs ∪ seed_seqs;
```

```
1 bps_reactivities_rep0 = shape_data_rep0.shape_reactivities[bps, nat_seqs,  
conds_sam_rep0];
```

```
1 nps_reactivities_rep0 = shape_data_rep0.shape_reactivities[nps, nat_seqs,  
conds_sam_rep0];
```

```
1 all_reactivities_rep0 = shape_data_rep0.shape_reactivities[:, nat_seqs, conds_sam_rep0];
```

```
1 shape_stats_rep0 = SamApp2024.shape_basepair_log_odds_v4(;  
2   shape_data = shape_data_rep0,  
3   paired_reactivities = bps_reactivities_rep0,  
4   unpaired_reactivities = nps_reactivities_rep0,  
5   all_reactivities = all_reactivities_rep0,  
6   only_hq_profile = true, p_thresh = 1e-3, nsamples = 1000  
7 );
```

```
_thresh = 1.6094379124341003
```

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

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

```
x_mg_rep0 =
```

```
▶ [-4.86644, -12.1404, -14.8342, -2.11477, 0.0, 0.0, -14.6983, 0.0, -4.65234, 0.0, -12.9099, -9.78
```

```
1 x_mg_rep0 = nansum(shape_stats_rep0.shape_log_odds[_sites, :, conds_mg_rep0]; dim=(1,3))
```

```
x_sam_rep0 =
```

```
▶ [15.7896, 4.08177, -18.098, 17.7391, 0.0, 0.0, 7.75366, 0.0, 13.8029, 0.0, -27.6326, 3.38205, 20
```

```
1 x_sam_rep0 = nansum(shape_stats_rep0.shape_log_odds[_sites, :, conds_sam_rep0]; dim=(1,3))
```

```
1 _responds_sam_yes_rep0 = (x_mg_rep0 .< -_thresh) .& (x_sam_rep0 .> +_thresh);
```

```
1 _responds_sam_nop_rep0 = (x_mg_rep0 .> +_thresh) .| (x_sam_rep0 .< -_thresh);
```

```
1 _inconclusive_rep0 = ((!).(_responds_sam_yes_rep0)) .& ((!).(_responds_sam_nop_rep0));
```

```
1 _conclusive_rep0 = _responds_sam_yes_rep0 .| _responds_sam_nop_rep0;
```

```
1 aptamer_rbm_energies = [  
2     ismissing(seq) ? missing :  
3     free_energy(SamApp2024.rbm2022(), SamApp2024.onehot(LongRNA{4}(seq)))  
4     for seq in shape_data_045.aligned_sequences  
5 ];
```

```
wuss =
```

```
"(((((((, , , , <<<<----<<<_AAAA>>>----->>>><<<<-----<<<<_>>>>----->>>>, , , , <aaaa<<<<----->>>>))
```

```
1 wuss = SamApp2024.rfam_ss("RF00162"; inserts=false)
```

```
ss =
```

```
"(((((((.....((((.....((((.....)))).....))))((((.....((((.....)))).....)))).....((((.....)))).....))))
```

```
1 ss = SamApp2024.clean_wuss(wuss)
```

```
1 p1_pos = SamApp2024.RF00162_sites_annotated_secondary_structure().p1;
```

```
1 p2_pos = SamApp2024.RF00162_sites_annotated_secondary_structure().p2;
```

```
1 p3_pos = SamApp2024.RF00162_sites_annotated_secondary_structure().p3;
```

```
1 p4_pos = SamApp2024.RF00162_sites_annotated_secondary_structure().p4;
```

```
1 pk_pos = SamApp2024.RF00162_sites_annotated_secondary_structure().pk;
```

```
1 ss_without_P1 = join([i ∈ p1_pos ? '.' : c for (i,c) in enumerate(ss)]);
```

```
1 ss_without_P2 = join([i ∈ p2_pos ? '.' : c for (i,c) in enumerate(ss)]);
```

```
1 ss_without_P3 = join([i ∈ p3_pos ? '.' : c for (i,c) in enumerate(ss)]);
```

```
1 ss_without_P4 = join([i ∈ p4_pos ? '.' : c for (i,c) in enumerate(ss)]);
```

```
ss_pk_only =  
".....((((.....)))).
```

```
1 ss_pk_only = replace(wuss, r"\(|\\|\\[|\\|\\{|\\}|\\<|\\>|\\-|\\-|\\," => '.', 'A' => '(', 'a'  
=> ')')
```

```
1 sampled_v = SamApp2024.rbm2022samples();
```

```
1 Vienna_energies_fold = [ismissing(seq) ? missing : ustrip(ViennaRNA.energy(string(seq),  
ss)) for seq = shape_data_rep0.aligned_sequences];
```

```
1 Vienna_energies_P1 = [ismissing(seq) ? missing : ustrip(ViennaRNA.energy(string(seq),  
ss)) - ustrip(ViennaRNA.energy(string(seq), ss_without_P1)) for seq =  
shape_data_rep0.aligned_sequences];
```

```
1 Vienna_energies_P2 = [ismissing(seq) ? missing : ustrip(ViennaRNA.energy(string(seq),  
ss)) - ustrip(ViennaRNA.energy(string(seq), ss_without_P2)) for seq =  
shape_data_rep0.aligned_sequences];
```

```
1 Vienna_energies_P3 = [ismissing(seq) ? missing : ustrip(ViennaRNA.energy(string(seq),  
ss)) - ustrip(ViennaRNA.energy(string(seq), ss_without_P3)) for seq =  
shape_data_rep0.aligned_sequences];
```

```
1 Vienna_energies_P4 = [ismissing(seq) ? missing : ustrip(ViennaRNA.energy(string(seq),  
ss)) - ustrip(ViennaRNA.energy(string(seq), ss_without_P4)) for seq =  
shape_data_rep0.aligned_sequences];
```

```
1 Vienna_energies_Pk = [ismissing(seq) ? missing : ustrip(ViennaRNA.energy(string(seq),  
ss_pk_only)) for seq = shape_data_rep0.aligned_sequences];
```

```
1 @time Vienna_energies_P1_RBM_samples = [  
2     ustrip(ViennaRNA.energy(string(seq), ss)) - ustrip(ViennaRNA.energy(string(seq),  
ss_without_P1))  
3     for seq = SamApp2024.rnaseq(sampled_v)  
4 ];
```

```
> 1.930079 seconds (2.18 M allocations: 129.627 MiB, 1.66% gc time, 49.02% compi  
lation time)
```

```
1 Vienna_energies_Pk_RBM_samples = [ustrip(ViennaRNA.energy(string(seq), ss_pk_only)) for  
seq = SamApp2024.rnaseq(sampled_v)];
```

```
Vienna_energies_Pk_RNAeval =
```

```
► [-6.6, -7.5, -6.0, -5.5, -8.1, NaN, -6.3, -5.1, -6.0, 2.9, -4.2, -5.5, -7.9, NaN, -7.5, 3.9, -6.3
```

```
1 Vienna_energies_Pk_RNAeval = [ismissing(seq) ? NaN :  
  SamApp2024.vienna_pk_binding_energy_rnaeval(seq) for seq =  
  shape_data_rep0.aligned_sequences]
```

```
>
```

```
WARNING: bases 25 and 80 (UU) can't pair!  
WARNING: bases 25 and 80 (AA) can't pair!  
WARNING: bases 27 and 78 (UC) can't pair!  
WARNING: bases 28 and 77 (GA) can't pair!  
WARNING: bases 25 and 80 (AG) can't pair!  
WARNING: bases 28 and 77 (AC) can't pair!  
WARNING: bases 25 and 80 (C-) can't pair!  
WARNING: bases 26 and 79 (U-) can't pair!  
WARNING: bases 27 and 78 (G-) can't pair!  
WARNING: bases 28 and 77 (G-) can't pair!  
WARNING: bases 26 and 79 (GA) can't pair!  
WARNING: bases 27 and 78 (GA) can't pair!  
WARNING: bases 25 and 80 (UU) can't pair!  
WARNING: bases 26 and 79 (AA) can't pair!  
WARNING: bases 25 and 80 (CU) can't pair!  
WARNING: bases 25 and 80 (AA) can't pair!  
WARNING: bases 25 and 80 (AG) can't pair!  
WARNING: bases 26 and 79 (AA) can't pair!  
WARNING: bases 27 and 78 (AA) can't pair!  
WARNING: bases 25 and 80 (A-) can't pair!  
WARNING: bases 25 and 80 (AG) can't pair!  
WARNING: bases 26 and 79 (AG) can't pair!  
WARNING: bases 27 and 78 (GA) can't pair!  
WARNING: bases 27 and 78 (GA) can't pair!  
WARNING: bases 25 and 80 (AA) can't pair!  
WARNING: bases 26 and 79 (UU) can't pair!  
WARNING: bases 25 and 80 (AG) can't pair!  
WARNING: bases 25 and 80 (CA) can't pair!  
WARNING: bases 26 and 79 (UU) can't pair!  
WARNING: bases 26 and 79 (UU) can't pair!  
WARNING: bases 26 and 79 (UU) can't pair!  
WARNING: bases 25 and 80 (AG) can't pair!  
WARNING: bases 25 and 80 (UU) can't pair!  
WARNING: bases 26 and 79 (AA) can't pair!  
WARNING: bases 28 and 77 (AC) can't pair!
```



```
1 Vienna_energies_Pk_RBM_samples_RNAeval =
  [SamApp2024.vienna_pk_binding_energy_rnaeval(string(seq)) for seq =
    SamApp2024.rnaseq(sampled_v)];
```

```
>
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 27 and 78 (GA) can't pair!
WARNING: bases 26 and 79 (U-) can't pair!
WARNING: bases 25 and 80 (AG) can't pair!
WARNING: bases 25 and 80 (AA) can't pair!
WARNING: bases 25 and 80 (CA) can't pair!
WARNING: bases 28 and 77 (GG) can't pair!
WARNING: bases 25 and 80 (UU) can't pair!
WARNING: bases 25 and 80 (UU) can't pair!
WARNING: bases 25 and 80 (AG) can't pair!
WARNING: bases 25 and 80 (AG) can't pair!
WARNING: bases 25 and 80 (CA) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 28 and 77 (AC) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 28 and 77 (AC) can't pair!
WARNING: bases 25 and 80 (AG) can't pair!
WARNING: bases 25 and 80 (--) can't pair!
WARNING: bases 27 and 78 (-C) can't pair!
WARNING: bases 28 and 77 (-U) can't pair!
WARNING: bases 25 and 80 (AG) can't pair!
WARNING: bases 27 and 78 (GA) can't pair!
WARNING: bases 25 and 80 (AG) can't pair!
WARNING: bases 27 and 78 (AC) can't pair!
WARNING: bases 25 and 80 (AA) can't pair!
WARNING: bases 26 and 79 (UU) can't pair!
WARNING: bases 25 and 80 (GA) can't pair!
WARNING: bases 26 and 79 (CU) can't pair!
WARNING: bases 25 and 80 (AA) can't pair!
WARNING: bases 25 and 80 (CA) can't pair!
WARNING: bases 28 and 77 (AC) can't pair!
```

```
1 # All merged data, for the reactivity profiles plots
2 shape_data_all_merged =
  SamApp2024.load_shapemapper_data_pierre_demux_20231027_repls_merged();
```

```
>
Downloading artifact: SAMAP_ALL-REP-MERGED-2023-10-27
Failure artifact: SAMAP_ALL-REP-MERGED-2023-10-27
Downloading artifact: SAMAP_ALL-REP-MERGED-2023-10-27
```

```
1 conds_SAM_all_merged = map(identity, indexin(["SAMAP_1M7_0-1SAM_5Mg_T30C_allrep",
  "SAMAP_1M7_1SAM_5Mg_T30C_allrep"], shape_data_all_merged.conditions));
```

```
1 conds_Mg_all_merged = map(identity, indexin(["SAMAP_1M7_noSAM_5Mg_T30C_allrep"],
  shape_data_all_merged.conditions));
```

```
► ([1, 2], [3])
```

```
1 conds_SAM_all_merged, conds_Mg_all_merged
```

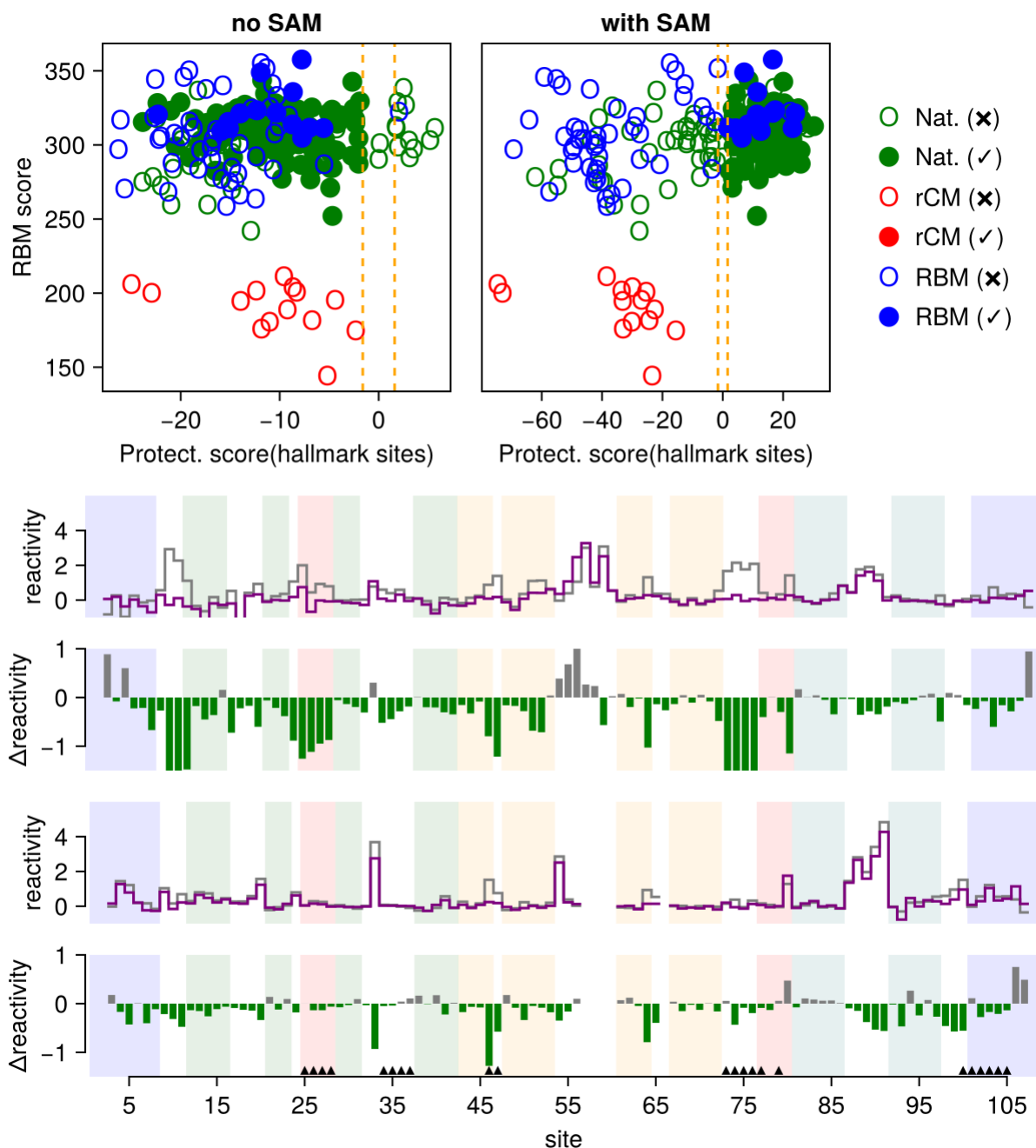
Multiple expressions in one cell.

How would you like to fix it?

- Split this cell into 4 cells, or
- Wrap all code in a `begin ... end` block.

```
1 bps_reactivities_merged = shape_data_all_merged.shape_reactivities[bps, nat_seqs,  
  conds_SAM_all_merged];  
2 nps_reactivities_merged = shape_data_all_merged.shape_reactivities[nps, nat_seqs,  
  conds_SAM_all_merged];  
3 all_reactivities_merged = shape_data_all_merged.shape_reactivities[:, nat_seqs,  
  conds_SAM_all_merged];  
4  
5 shape_stats_merged = SamApp.shape_basepair_log_odds_v4(  
6   shape_data = shape_data_all_merged,  
7   paired_reactivities = bps_reactivities_merged,  
8   unpaired_reactivities = nps_reactivities_merged,  
9   all_reactivities = all_reactivities_merged,  
10  only_hq_profile = true, p_thresh = 1e-3, nsamples = 1000  
11 );
```

```
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(; halign = :left)
2 ax = Makie.Axis(fig[1,1][1,1]; halign=:left, width=200, height=200, xlabel="Protect.
  score(hallmark sites)", ylabel="RBM score", title="no SAM", xgridvisible=false,
  ygridvisible=false)
3 #Makie.scatter!(ax, x_mg_rep0, -aptamer_rbm_energies, markersize=10, color=(silver,
  0.3), label="All probed")
4 Makie.scatter!(ax, x_mg_rep0[findall(_responds_sam_nop_rep0) ∩ nat_seqs], -
  aptamer_rbm_energies[findall(_responds_sam_nop_rep0) ∩ nat_seqs], markersize=15,
  color=:green, marker='o')
5 Makie.scatter!(ax, x_mg_rep0[findall(_responds_sam_yes_rep0) ∩ nat_seqs], -
  aptamer_rbm_energies[findall(_responds_sam_yes_rep0) ∩ nat_seqs], markersize=15,
  color=:green, marker='●')
6 Makie.scatter!(ax, x_mg_rep0[findall(_responds_sam_nop_rep0) ∩ inf_seqs], -
  aptamer_rbm_energies[findall(_responds_sam_nop_rep0) ∩ inf_seqs], markersize=15,
  color=:red, marker='o')
7 Makie.scatter!(ax, x_mg_rep0[findall(_responds_sam_yes_rep0) ∩ inf_seqs], -
  aptamer_rbm_energies[findall(_responds_sam_yes_rep0) ∩ inf_seqs], markersize=15,
  color=:red, marker='●')
8 Makie.scatter!(ax, x_mg_rep0[findall(_responds_sam_nop_rep0) ∩ rbm_seqs], -
  aptamer_rbm_energies[findall(_responds_sam_nop_rep0) ∩ rbm_seqs], markersize=15,
  color=:blue, marker='o')

```

```

9 Makie.scatter!(ax, x_mg_rep0[findall(_responds_sam_yes_rep0) ∩ rbm_seqs], -
aptamer_rbm_energies[findall(_responds_sam_yes_rep0) ∩ rbm_seqs], markersize=15,
color=:blue, marker='●')
10 Makie.vlines!(ax, [-_thresh, _thresh], linestyle=:dash, color=:orange)
11
12 ax = Makie.Axis(fig[1,1][1,2]; halight=:left, width=200, height=200, xlabel="Protect.
score(hallmark sites)", ylabel="RBM score", title="with SAM", xgridvisible=false,
ygridvisible=false)
13 #Makie.scatter!(ax, x_sam_rep0, -aptamer_rbm_energies, markersize=10, color=(gray,
0.5), label="All probed")
14 plt1 = Makie.scatter!(ax, x_sam_rep0[findall(_responds_sam_nop_rep0) ∩ nat_seqs], -
aptamer_rbm_energies[findall(_responds_sam_nop_rep0) ∩ nat_seqs], markersize=15,
color=:green, marker='O', label="Nat. (×)")
15 plt2 = Makie.scatter!(ax, x_sam_rep0[findall(_responds_sam_yes_rep0) ∩ nat_seqs], -
aptamer_rbm_energies[findall(_responds_sam_yes_rep0) ∩ nat_seqs], markersize=15,
color=:green, marker='●', label="Nat. (✓)")
16 plt3 = Makie.scatter!(ax, x_sam_rep0[findall(_responds_sam_nop_rep0) ∩ inf_seqs], -
aptamer_rbm_energies[findall(_responds_sam_nop_rep0) ∩ inf_seqs], markersize=15,
color=:red, marker='O', label="CM (×)")
17 plt4 = Makie.scatter!(ax, x_sam_rep0[findall(_responds_sam_yes_rep0) ∩ inf_seqs], -
aptamer_rbm_energies[findall(_responds_sam_yes_rep0) ∩ inf_seqs], markersize=15,
color=:red, marker='●', label="CM (✓)")
18 plt5 = Makie.scatter!(ax, x_sam_rep0[findall(_responds_sam_nop_rep0) ∩ rbm_seqs], -
aptamer_rbm_energies[findall(_responds_sam_nop_rep0) ∩ rbm_seqs], markersize=15,
color=:blue, marker='O', label="RBM (×)")
19 plt6 = Makie.scatter!(ax, x_sam_rep0[findall(_responds_sam_yes_rep0) ∩ rbm_seqs], -
aptamer_rbm_energies[findall(_responds_sam_yes_rep0) ∩ rbm_seqs], markersize=15,
color=:blue, marker='●', label="RBM (✓)")
20 Makie.vlines!(ax, [-_thresh, _thresh], linestyle=:dash, color=:orange)
21 #Makie.xlims!(ax, -78, 35)
22 Makie.hideydecorations!(ax)
23 #Makie.axislegend(ax, position=:rt, framevisible=false)
24 Makie.Legend(fig[1,1][1,3],
25     [plt1, plt2, plt3, plt4, plt5, plt6],
26     ["Nat. (×)", "Nat. (✓)", "rCM (×)", "rCM (✓)", "RBM (×)", "RBM (✓)"],
27     framevisible=false
28 )
29
30 # _dummy_ax = Makie.Axis(fig[1,2], width=400, height=300, xgridvisible=false,
ygridvisible=false) # placeholder for the table
31 # Makie.hidexdecorations!(_dummy_ax)
32 # Makie.hideydecorations!(_dummy_ax)
33 # Makie.hidespines!(_dummy_ax, :t, :b, :l, :r)
34
35 n_ex_rbm = 299 # rbm example, switcher
36 #n_ex_nat = 112 # natural example, responsive but not switcher
37 n_ex_nat = 101 # natural example, responsive but not switcher
38
39 _width = 550
40 _height = 70
41
42 _R_sam = shape_data_all_merged.shape_reactivities[:, n_ex_rbm,
conds_SAM_all_merged[1]]
43 _R_mg = shape_data_all_merged.shape_reactivities[:, n_ex_rbm,
only(conds_Mg_all_merged)]
44
45 ax_react_1 = Makie.Axis(fig[2,1][1,1], width=_width, height=_height,
xticks=5:10:108, ylabel="reactivity", xgridvisible=false, ygridvisible=false,
yticks=0:2:5, ytrimspine=true)
46 for (x0, xf, color, alpha) = struct_bands

```

```

47     Makie.vspan!(ax_react_1, x0, xf; color=(color, alpha))
48 end
49 Makie.stairs!(ax_react_1, 1:108, _R_mg, color=:gray, step=:center)
50 Makie.stairs!(ax_react_1, 1:108, _R_sam, color=:purple, step=:center)
51 Makie.xlims!(ax_react_1, 0.5, 108.5)
52 Makie.ylims!(ax_react_1, -1, 4)
53 Makie.hidespines!(ax_react_1, :t, :r, :b)
54 Makie.hidexdecorations!(ax_react_1)
55
56 ax_diff_1 = Makie.Axis(fig[2,1][2,1]; width=_width, height=_height, xticks=5:10:108,
xlabel="site", ylabel="Δreactivity", xgridvisible=false, ygridvisible=false,
xtrimspine=true, ytrimspine=true, yticks=-2:2)
57 for (x0, xf, color, alpha) = struct_bands
58     Makie.vspan!(ax_diff_1, x0, xf; color=(color, alpha))
59 end
60 Makie.barplot!(ax_diff_1, 1:108, _R_sam - _R_mg, color=ifelse.(_R_sam - _R_mg .< 0,
:green, :gray))
61 Makie.xlims!(ax_diff_1, 0.5, 108.5)
62 Makie.ylims!(ax_diff_1, -1.5, 1.5)
63 Makie.hidespines!(ax_diff_1, :t, :b, :r)
64 Makie.hidexdecorations!(ax_diff_1)
65
66 _R_sam = shape_data_all_merged.shape_reactivities[:, n_ex_nat,
conds_SAM_all_merged[1]]
67 _R_mg = shape_data_all_merged.shape_reactivities[:, n_ex_nat,
only(conds_Mg_all_merged)]
68
69 ax_react_2 = Makie.Axis(fig[2,1][3,1], width=_width, height=_height,
xticks=10:10:108, ylabel="reactivity", xgridvisible=false, ygridvisible=false,
yticks=0:2:5, ytrimspine=true)
70 for (x0, xf, color, alpha) = struct_bands
71     Makie.vspan!(ax_react_2, x0, xf; color=(color, alpha))
72 end
73 Makie.stairs!(ax_react_2, 1:108, _R_mg, color=:gray, step=:center)
74 Makie.stairs!(ax_react_2, 1:108, _R_sam, color=:purple, step=:center)
75 Makie.xlims!(ax_react_2, 0.5, 108.5)
76 Makie.ylims!(ax_react_2, -1, 6)
77 Makie.hidespines!(ax_react_2, :t, :r, :b)
78 Makie.hidexdecorations!(ax_react_2)
79 #Makie.hideydecorations!(ax_react_2)
80
81 ax_diff_2 = Makie.Axis(fig[2,1][4,1]; width=_width, height=_height, xticks=5:10:108,
xlabel="site", ylabel="Δreactivity", xgridvisible=false, ygridvisible=false,
xtrimspine=true, ytrimspine=true, yticks=-2:2)
82 for (x0, xf, color, alpha) = struct_bands
83     Makie.vspan!(ax_diff_2, x0, xf; color=(color, alpha))
84 end
85 Makie.barplot!(ax_diff_2, 1:108, _R_sam - _R_mg, color=ifelse.(_R_sam - _R_mg .< 0,
:green, :gray))
86 Makie.scatter!(ax_diff_2, _sites, -1.4one.(_sites), markersize=7, color=:black,
marker=:utriangle)
87 Makie.xlims!(ax_diff_2, 0, 109)
88 Makie.ylims!(ax_diff_2, -1.5, 1)
89 Makie.hidespines!(ax_diff_2, :t, :r)
90 #Makie.hideydecorations!(ax_diff_2)
91
92 # Makie.scatter!(ax_diff_1, _sites, one.(_sites), color=:blue, markersize=5)
93 # Makie.scatter!(ax_diff_2, _sites, one.(_sites), color=:blue, markersize=5)
94
95 Makie.linkxaxes!(ax_react_1, ax_diff_1)

```

```
96 Makie.linkxaxes!(ax_react_2, ax_diff_2)
97 Makie.linkyaxes!(ax_react_1, ax_react_2)
98 Makie.linkyaxes!(ax_diff_1, ax_diff_2)
99
100 Makie.resize_to_layout!(fig)
101 #Makie.save("Figures/SAM response Rep10 v2.pdf", fig)
```