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

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

```julia
import CairoMakie
```

```julia
import CSV
```

```julia
import HDF5
```

```julia
import FASTX
```

```julia
import Infernal
```

```julia
import KernelDensity
```

```julia
import Makie
```

```julia
import RestrictedBoltzmannMachines as RBMs
```

```julia
import Rfam
```

```julia
import SamApp2024
```

```julia
import StatsBase
```

```julia
import Logomaker
```

```
    CondaPkg Found dependencies: /home/cossio/.julia/packages/Logomaker/8Rip9/CondaPkg.toml
    CondaPkg Found dependencies: /home/cossio/.julia/packages/PythonCall/S5MOg/CondaPkg.toml
    CondaPkg Dependencies already up to date
```

```julia
using BioSequences: LongRNA
```

```julia
using DataFrames: DataFrame
```

```julia
using Distributions: Gamma
```

```julia
using Distributions: logpdf
```

```julia
using Distributions: pdf
```

```julia
using Distributions: Poisson
```

```julia
using LinearAlgebra: Diagonal
using LinearAlgebra: eigen
```

```julia
using Makie: @L_str
```

```julia
using NaNStatistics: nansum
```

```julia
using Random: bitrand
```

```julia
using Statistics: cor
```

```julia
using Statistics: mean
```

```julia
using StatsBase: countmap
```

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

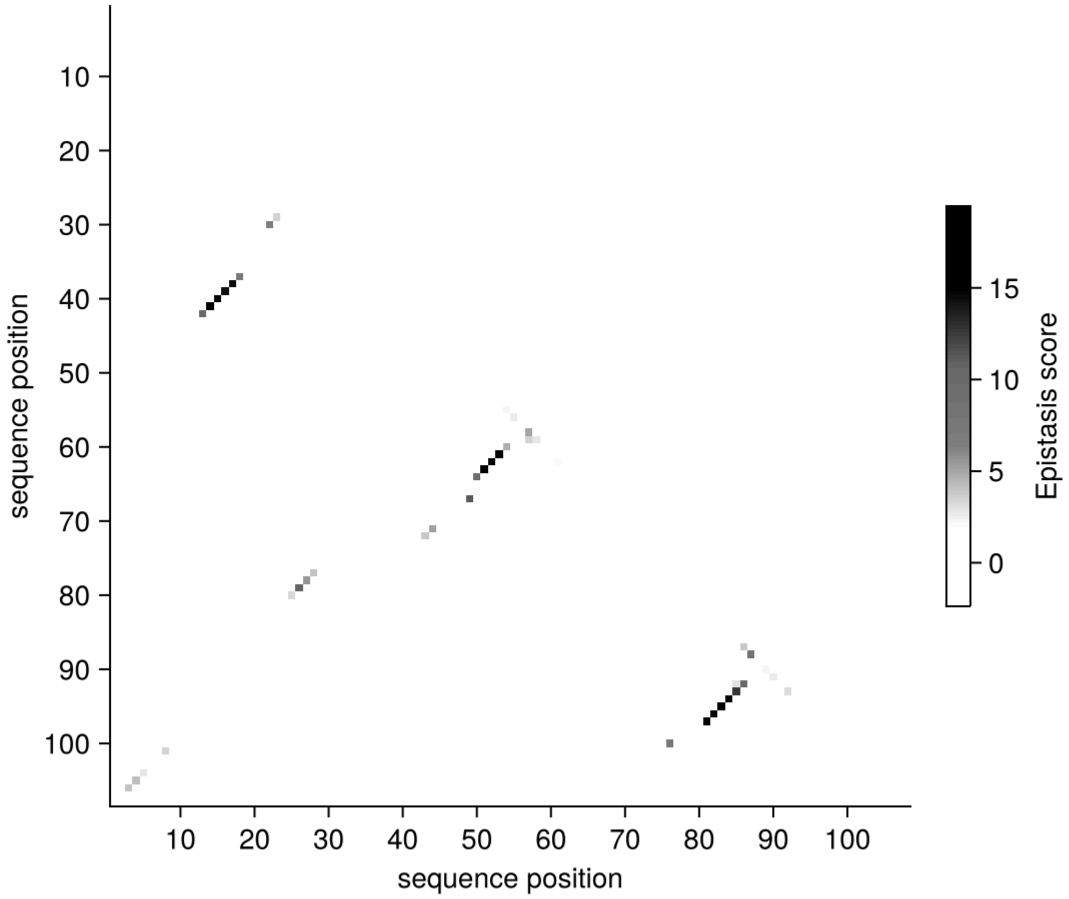
```julia
# trimmed (no inserts) aligned fasta
RF00162_hits_afa = Infernal.cmalign(Rfam_cm.out, Rfam.fasta_file("RF00162"); matchonly=true, outformat="AFA");
# these are already aligned and without inserts
```

```julia
RF00162_hits_sequences = LongRNA{4}.(FASTX.sequence.(FASTX.FASTA.Reader(open(RF00162_hits_afa.out))));
```

```julia
begin
    # APC-corrected contact map
    Cmap = SamApp2024.effective_contacts(SamApp2024.rbm2022(), SamApp2024.onehot(RF00162_hits_sequences));

    # zero-out diagonal
    for i in axes(Cmap, 1)
        Cmap[i,i] = 0
    end
end
```

```julia
begin
    plotted_matrix = copy(Cmap)
    for i = axes(plotted_matrix, 1), j = axes(plotted_matrix, 2)
        if i < j
            plotted_matrix[i,j] = Cmap[i,j]
        else
            plotted_matrix[i,j] = 0
        end
    end
end
```



```julia
let fig = Makie.Figure()
    ax = Makie.Axis(fig[1,1], yreversed=true, width=400, height=400, xticks=0:10:size(Cmap,1), yticks=0:10:size(Cmap,2), xlabel="sequence position", ylabel="sequence position")
    hm = Makie.heatmap!(ax, plotted_matrix, colormap=["white", "white", "gray", "dimgray", "black", "black"])
    Makie.hidespines!(ax, :t, :r)
    Makie.Colorbar(fig[1,2], hm, height=200, label="Epistasis score")
    Makie.resize_to_layout!(fig)
    #Makie.save("Figures/Contacts bw.pdf", fig)
    fig
end
```