BioMakie.jl

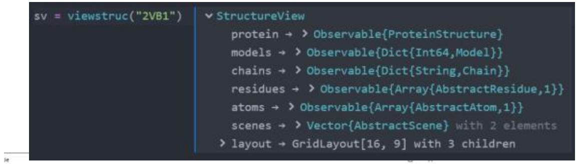
Daniel Kool, Jernigan Lab, ISU
GitHub: kool7d, Email: kool@iastate.edu

About:

BioMakie.jl is a package for biological visualization using Makie. The intent of this package is to facilitate visualization and connection of biological data. Makie is an ideal library for exploration of data representation and provides tools that make the handling and synchronization of events tied to data much simpler.

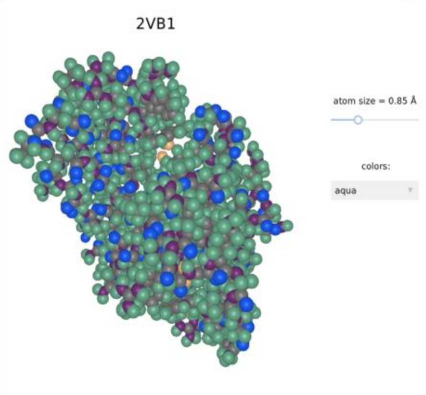
Protein Structure

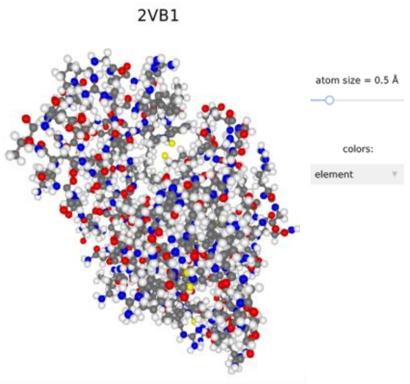
To view a PDB structure, use the `**viewstruc**` function with a PDB ID. The following code plots the structure then returns a *StructureView* object with relevant Nodes/Observables, the scene, and the layout for convenience.



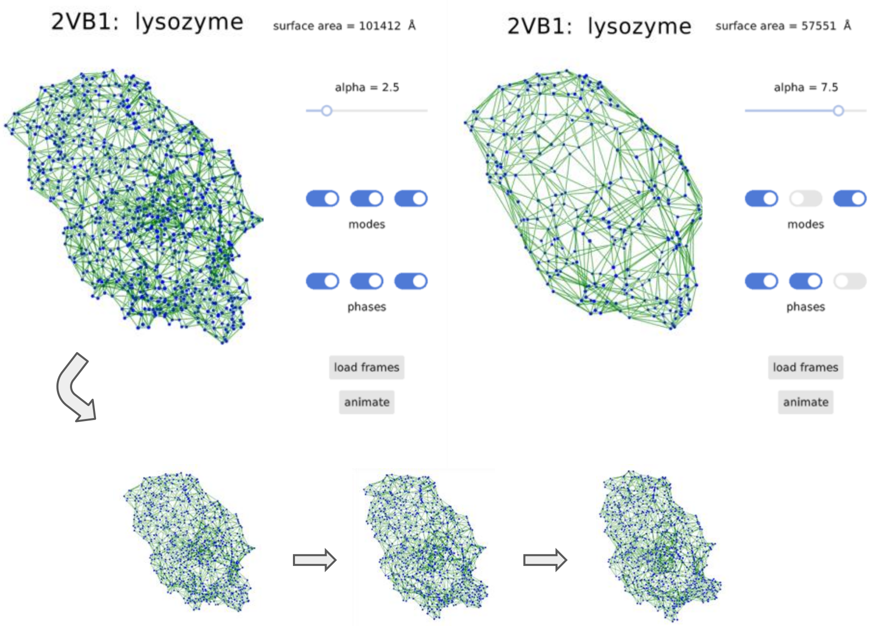
```
sv = viewstruc("2VB1")    ⌄ StructureView
                               protein → › Observable{ProteinStructure}
                               models → › Observable{Dict{Int64,Model}}
                               chains → › Observable{Dict{String,Chain}}
                               residues → › Observable{Array{AbstractResidue,1}}
                               atoms → › Observable{Array{AbstractAtom,1}}
                               scenes → › Vector{AbstractScene} with 2 elements
                             › layout → GridLayout[16, 9] with 3 children
```

The `**viewstruc**` function uses BioStructures.jl to retrieve a structure file from the Protein Data Bank (PDB). The figure above shows what it looks like in the Juno IDE.





The figures below show mesh representations of the lysozyme protein using the alpha shape algorithm[1]. Changing the alpha value produces a different mesh, and the surface area is calculated dynamically. This example also can be animated, showing motion along elastic network model normal modes using the NOLB method[2]. The view and the alpha value can be changed during the animation. The mesh is also updated throughout the animations. (vertices and edges change)



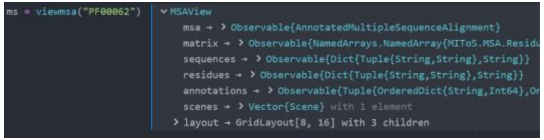
BioMakie.jl

Daniel Kool, Jernigan Lab, ISU
GitHub: kool7d, Email: kool@iastate.edu

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Multiple Sequence Alignment

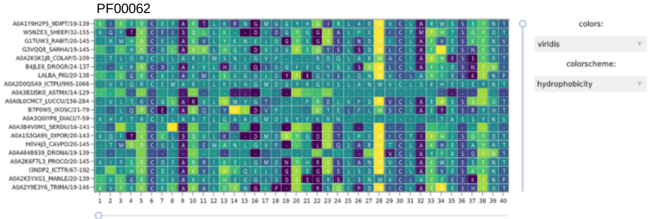
To view a multiple sequence alignment (MSA), use the `**viewmsa**` function with a Pfam ID. The following code plots the MSA then returns a *MSAView* object, similar to *StructureView*.

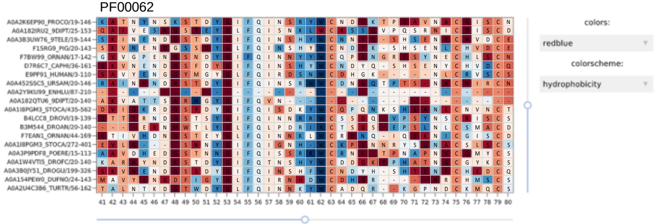


```
ms = viewmsa("PF00062")
```

```
⌄ MSAView
    msa → › Observable{AnnotatedMultipleSequenceAlignment}
    matrix → › Observable{NamedArrays.NamedArray{MIToS.MSA.Residu
    sequences → › Observable{Dict{Tuple{String,String},String}}
    residues → › Observable{Dict{Tuple{String,String},String}}
    annotations → › Observable{Tuple{OrderedDict{String,Int64},Or
    scenes → › Vector{Scene} with 1 element
  › layout → GridLayout[8, 16] with 3 children
```

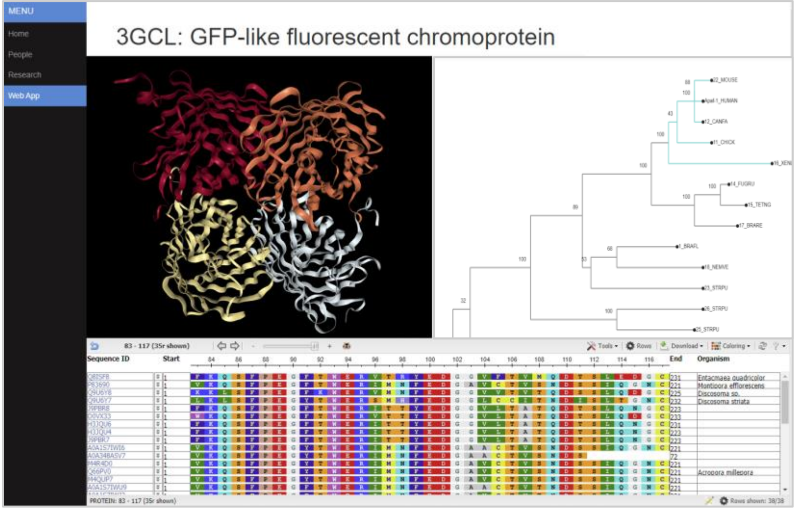
The `**viewmsa**` function uses MIToS.jl to retrieve an MSA file from the Protein family database (Pfam).





Current/Future Work

- Phylogenetics / Phylo trees
- More options and features everywhere
- Structural and dynamical analysis using packages such as Flux, DifferentialEquations, Flux3D
- Inclusion of work by other members of the Jernigan Lab such as new MSA and hinge-prediction methods
- Integration with JavaScript elements and web pages using WGLMakie and JSServe (example below)



Franklin.jl is being used in this example for serving the site, NGL.js for structure, Cytoscape.js for phylogeny tree, and NCBI's MSA viewer.

References:
1. https://en.wikipedia.org/wiki/Alpha_shape
2. Alexandre Hoffmann & Sergei Grudinin. NOLB : Non-linear rigid block normal mode analysis method. Journal of Chemical Theory and Computation, 2017, 13 (5), pp.2123-2134. DOI: 10.1021/acs.jctc.7b00197