New layouts, data types, and architecture for GRNsight v3: a web application for visualizing gene regulatory networks

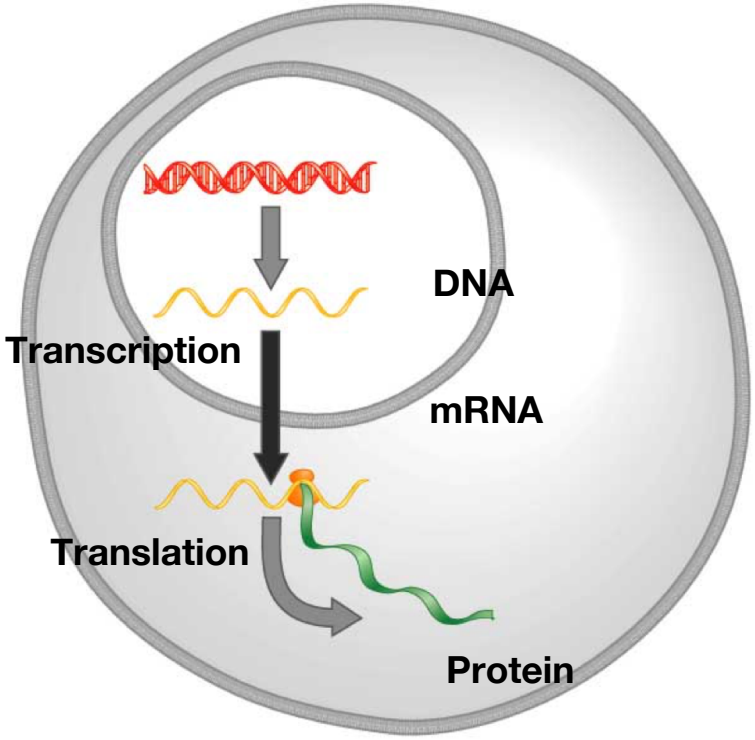
Mihir Samdarshi
Department of Biology
Loyola Marymount University

March 23, 2019
LMU Undergraduate Research Symposium

Outline

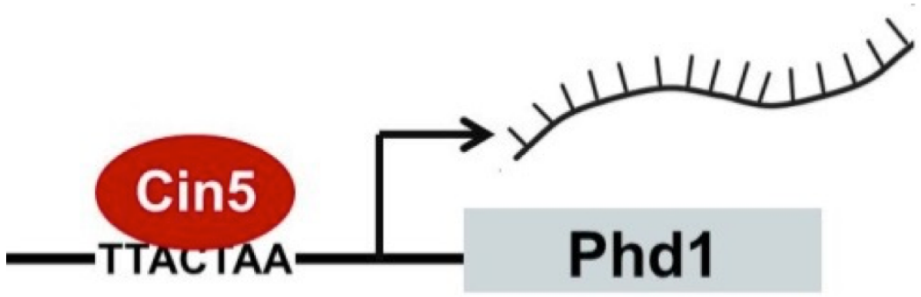
- Gene regulatory networks (GRNs) consist of genes, transcription factors, and the regulatory relationships between them.
- GRNsight is an open source web application and service for visualizing models of gene regulatory networks.
- GRNsight has matured since version 1.0, adding new features such as new layout options and data types.
- This has necessitated a complete refactor of GRNsight code to the Model-View-Controller paradigm, which facilitates easier user and developer interactions with the application.

Transcription factors regulate the expression of genes

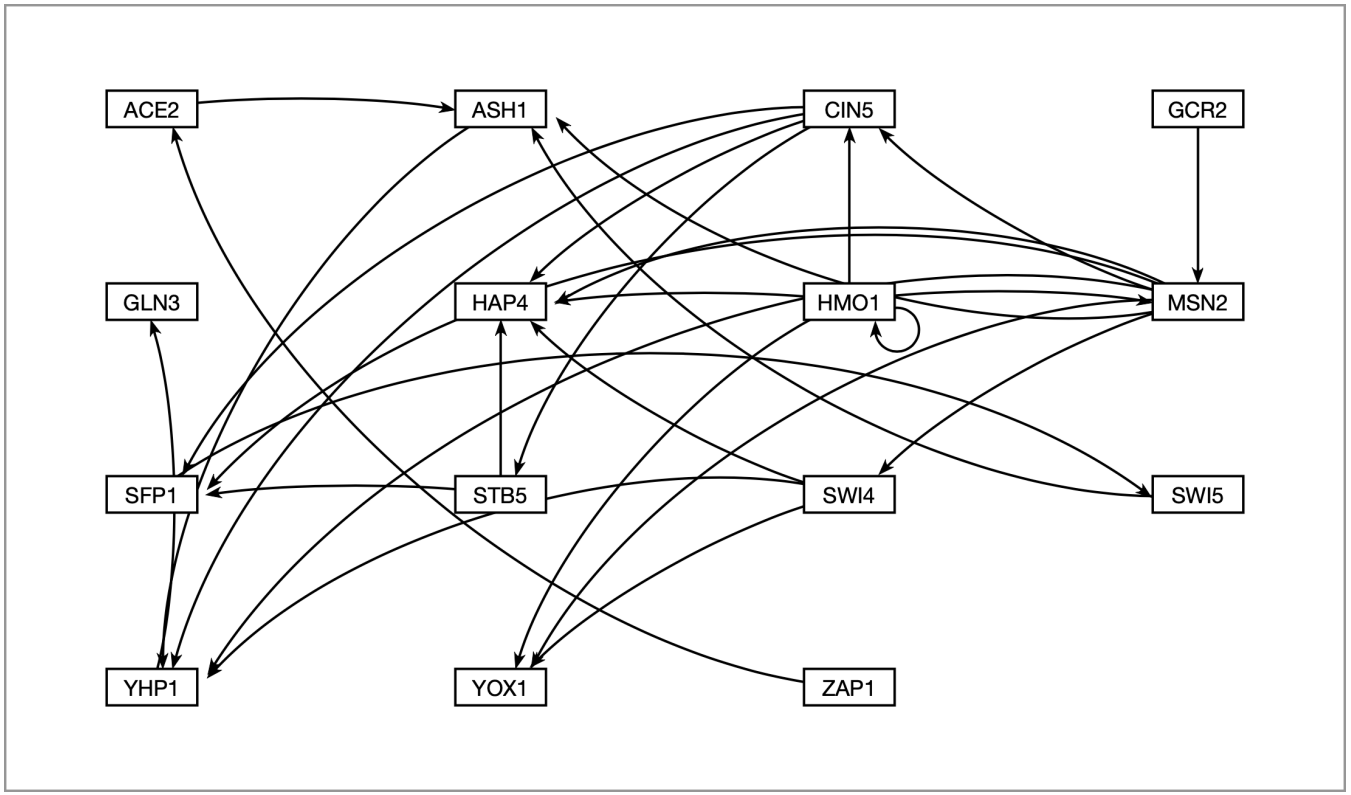


Gene regulatory process

- Transcription factors **activate** or **repress** expression of their target genes.



A gene regulatory network (GRN) controls the level of expression of genes and proteins



Dahlquist et al. (2015) *Bulletin of Mathematical Biology* 77: 1457.

GRNmap models the dynamics of GRNs to estimate weight parameters (*w*)

Magnitude of weights represent the influence of transcription factor on its target gene

Positive weights = **activation** of gene expression

Negative weights = **repression** of gene expression

$$\frac{dx_i(t)}{dt} = \frac{P_i}{1+\exp\left(-\left(\sum_j \left(w_{ij}x_j(t)\right) - b_i\right)\right)} - d_i x_i(t)$$

Expression patterns of each individual gene are modelled by individual differential equations

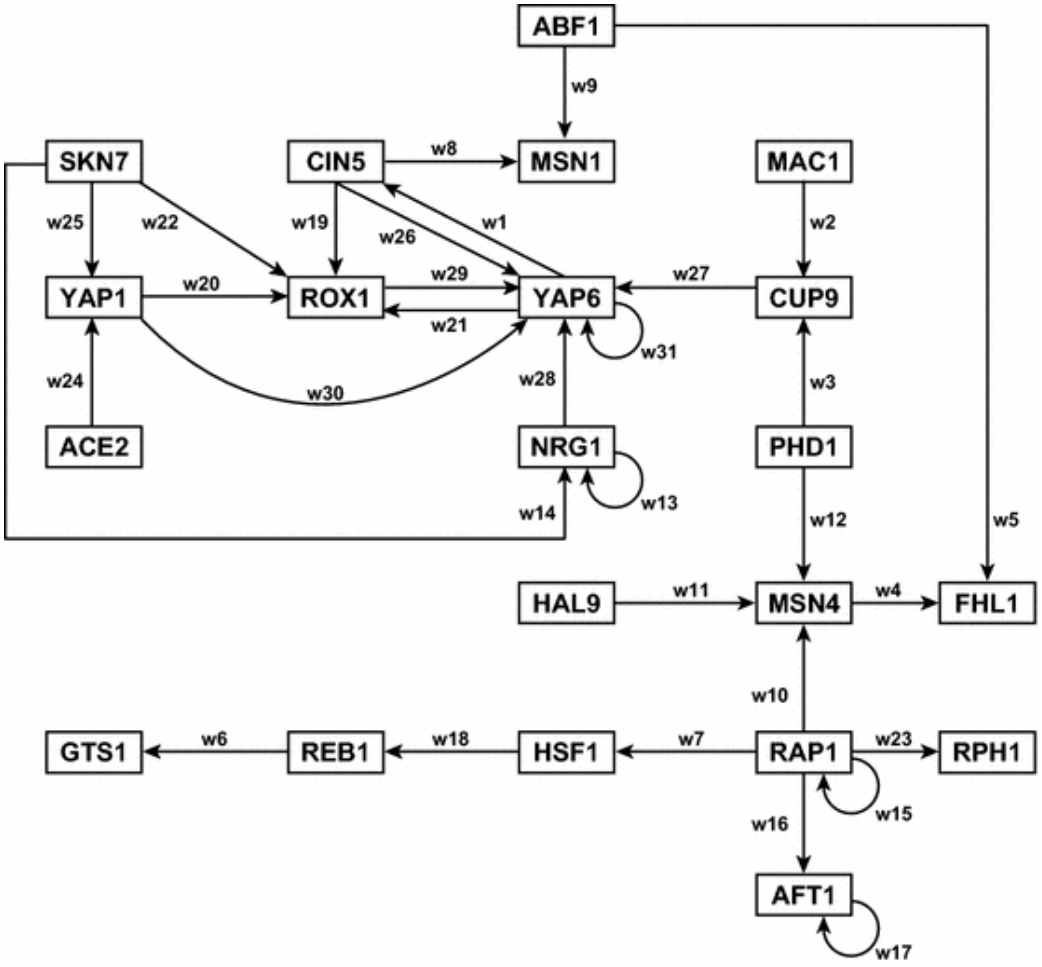
Weight values are outputted from GRNmap as an Excel spreadsheet

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	cols regulators/rows targets	ABF1	ACE2	AFT1	CIN5	CUP9	FHL1	GTS1	HAL9	HSF1	MAC1	MSN1	MSN4	NRG1	PHD1	RAP1	REB1	ROX1	RPH1	SKN7	YAP1	YAP6
2	ABF1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	ACE2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	AFT1	0	0	-0.8966	0	0	0	0	0	0	0	0	0	0	0	-0.4030	0	0	0	0	0	0
5	CIN5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.0450
6	CUP9	0	0	0	0	0	0	0	0	0	-0.1882	0	0	0	-0.6510	0	0	0	0	0	0	0
7	FHL1	0.1562	0	0	0	0	0	0	0	0	0	0	0.6121	0	0	0	0	0	0	0	0	0
8	GTS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0778	0	0	0	0	0
9	HAL9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	HSF1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1.2321	0	0	0	0	0	0
11	MAC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	MSN1	-2.9707	0	0	0.9393	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	MSN4	0	0	0	0	0	0	0	1.4283	0	0	0	0	0	0.5447	1.0131	0	0	0	0	0	0
14	NRG1	0	0	0	0	0	0	0	0	0	0	0	0	1.2341	0	0	0	0	0	-0.1852	0	0
15	PHD1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	RAP1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.8890	0	0	0	0	0	0
17	REB1	0	0	0	0	0	0	0	0	-0.0102	0	0	0	0	0	0	0	0	0	0	0	0
18	ROX1	0	0	0	-0.9278	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5744	-0.4315	-0.5071
19	RPH1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.4999	0	0	0	0	0	0
20	SKN7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	YAP1	0	-1.3615	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-0.4082	0	0
22	YAP6	0	0	0	-0.5312	-0.1293	0	0	0	0	0	0	0	0.6215	0	0	0	-0.7503	0	0	0.0146	-0.3027

.xlsx

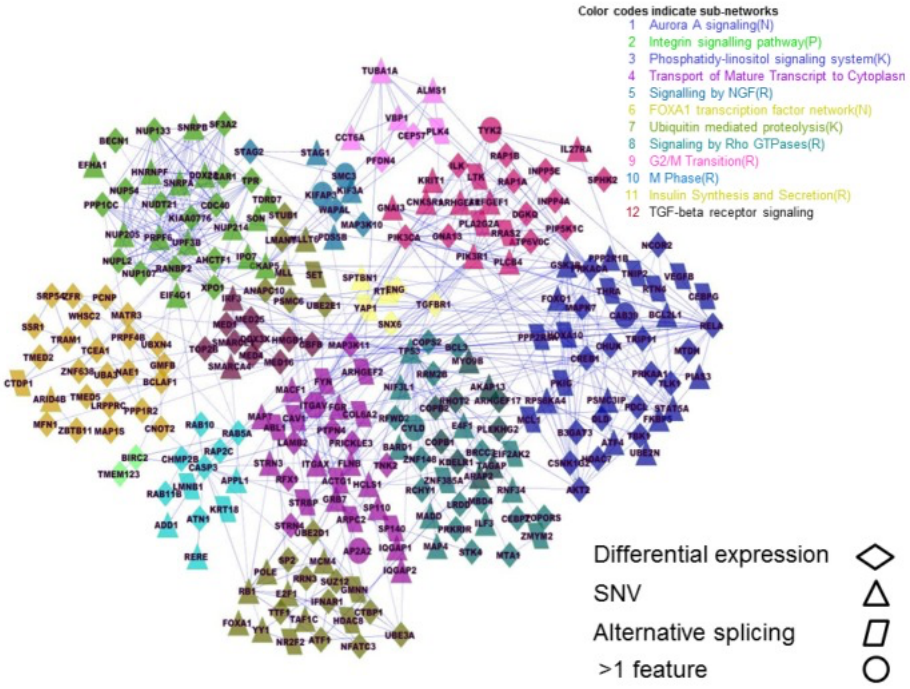
21 gene, 31 edge weighted network output from GRNmap

Before GRNsight, graphs would be manually plotted using Adobe Illustrator



Dahlquist et al. (2015) *Bulletin of Mathematical Biology* 77: 1457.

Cytoscrape and Gephi are examples of software for large graph visualizations



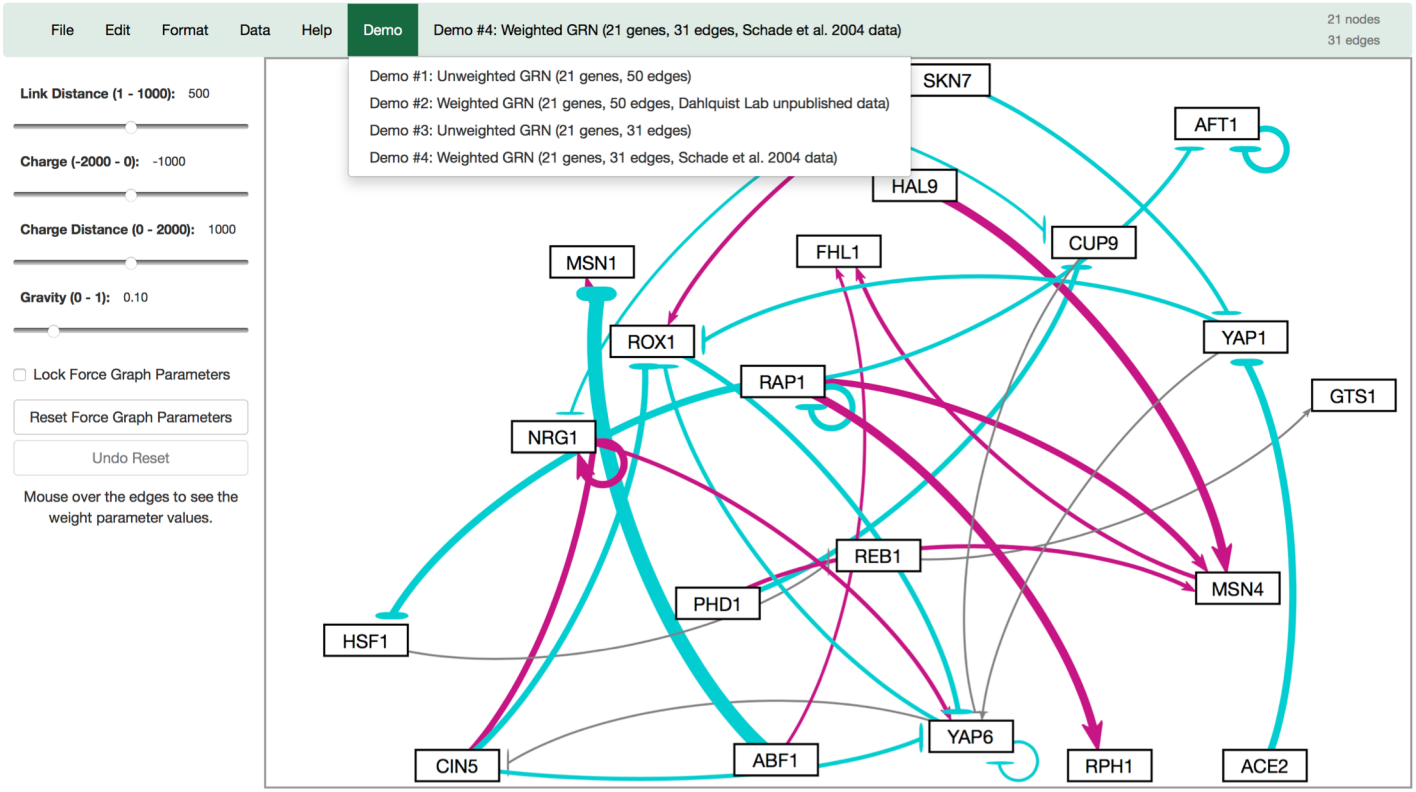
Great for overall topography
Not great for individual relationships
Emphasis on nodes

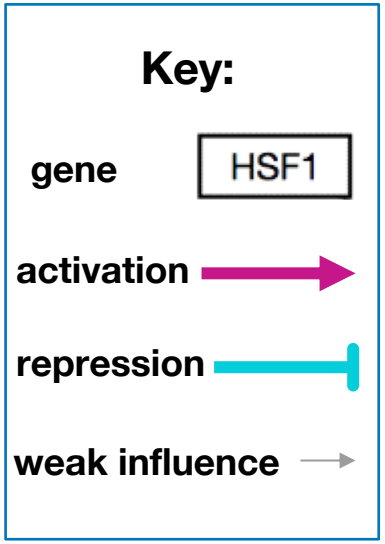
Karlari et al. (2013) *PLoS One* 8: E79398.
Poynter et al. (2015) BMC Cancer 15: 769.

GRNsight fills a niche in bioinformatics visualization software

Cytoscape and Gephi	GRNsight
Optimized for large-scale graphs.	✓ Optimized for small- to medium- scale graphs.
Must be installed.	✓ Exists as a web application.
Several complex features in addition to visualization.	✓ Easy to use, and easily interpretable visualizations.
Designed to emphasize node placement	✓ Designed to emphasize individual relationships

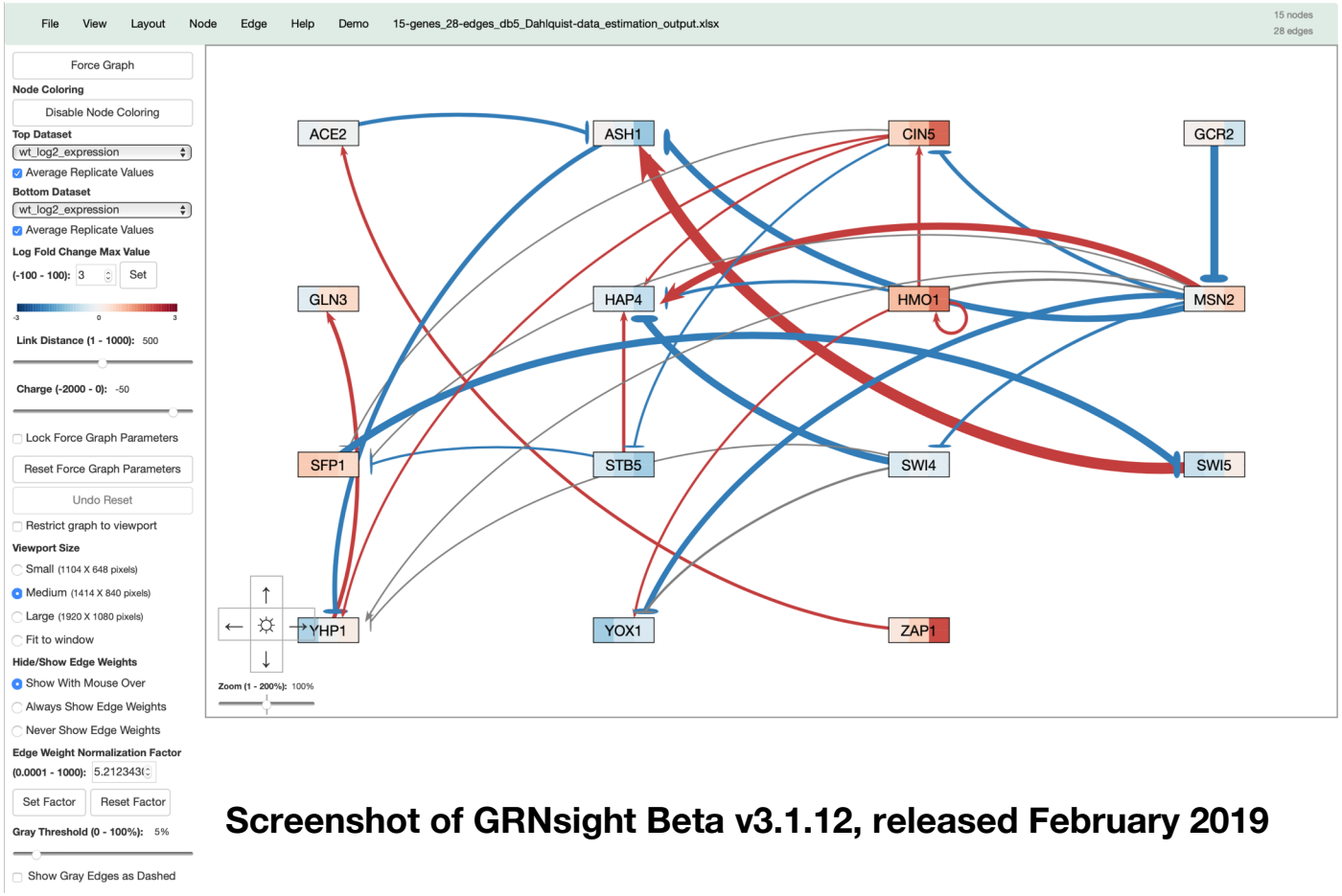
GRNsight automatically produces visualizations of weighted networks from adjacency matrix inputs



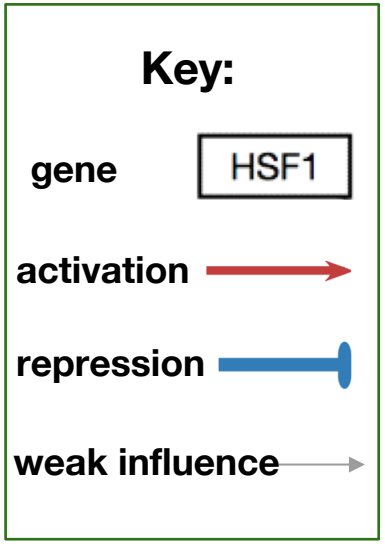


Screenshot of GRNsight v1, released Aug 2016

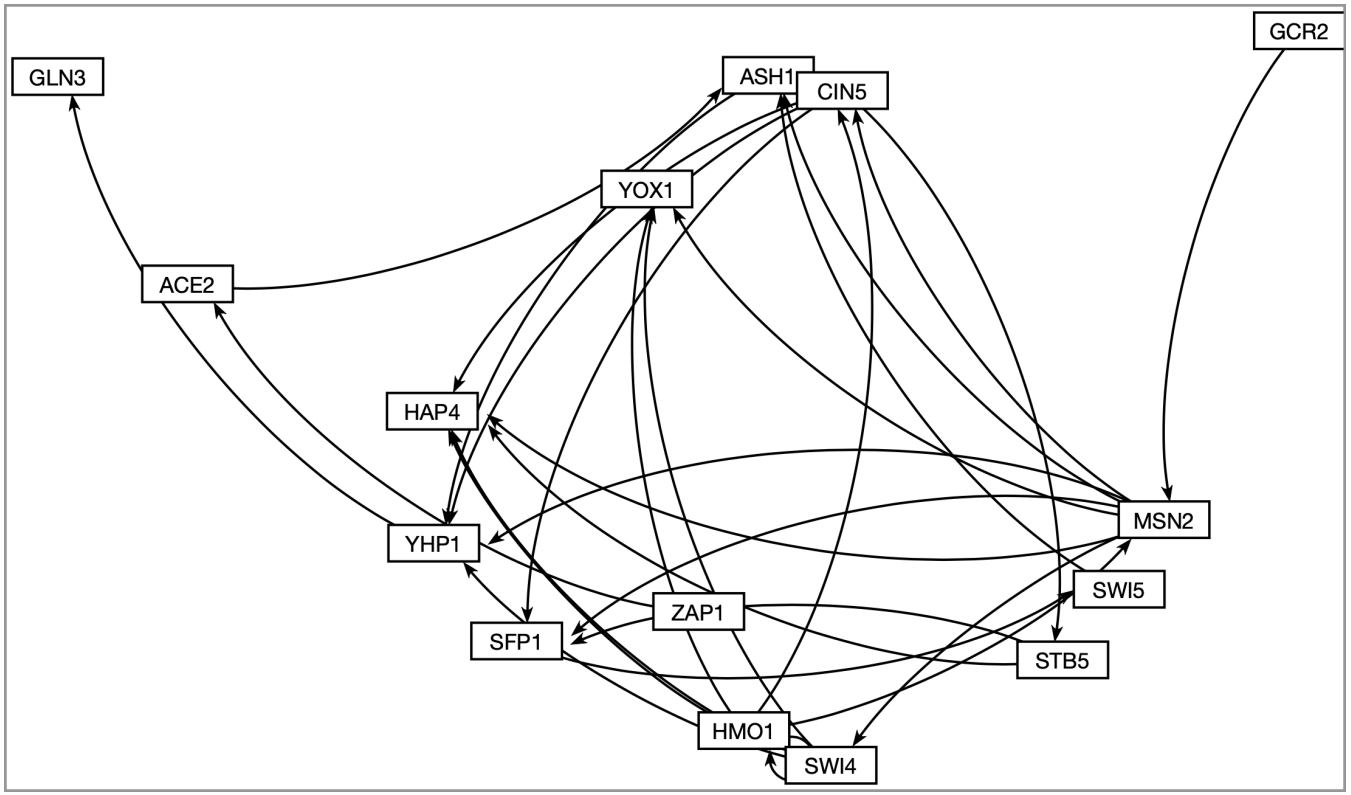
GRNsight has matured to version 3 with many new features added



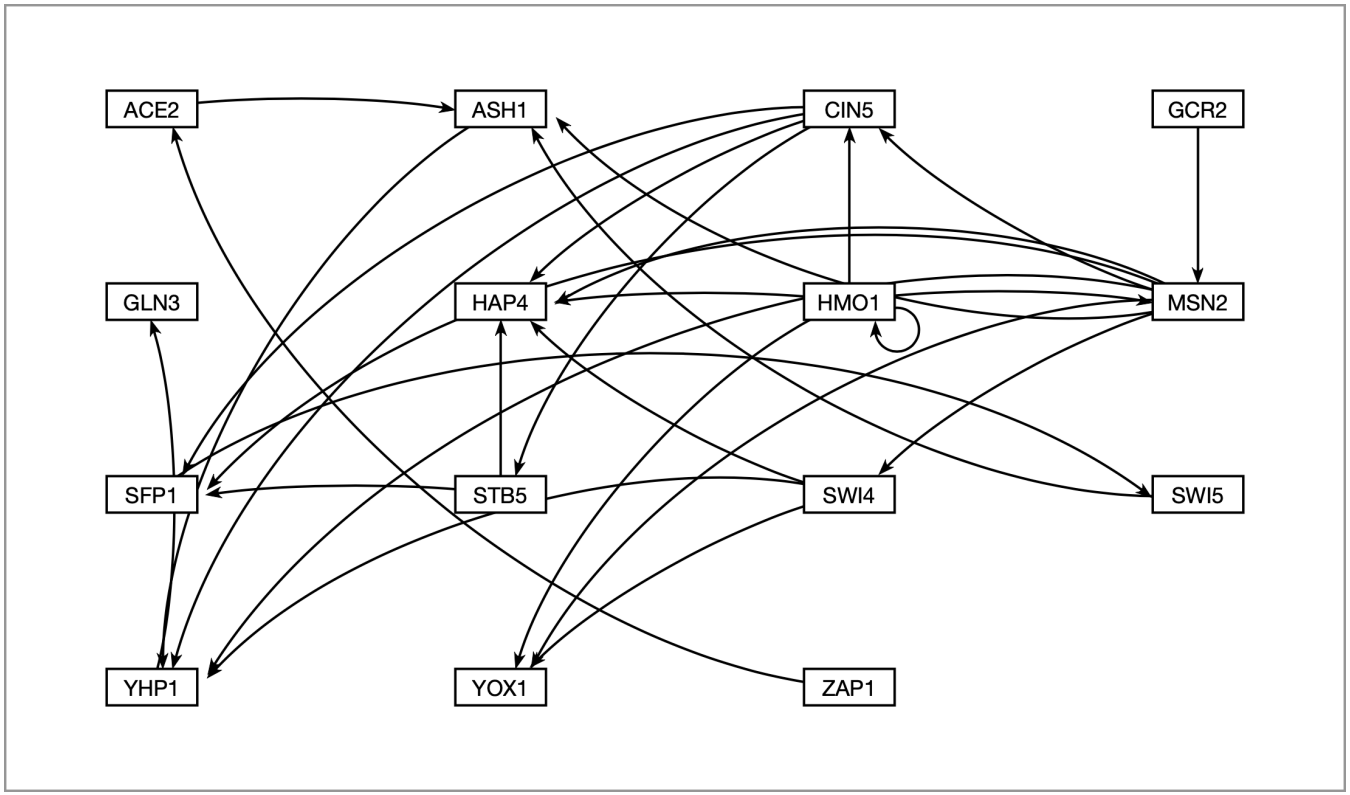
Screenshot of GRNsight Beta v3.1.12, released February 2019



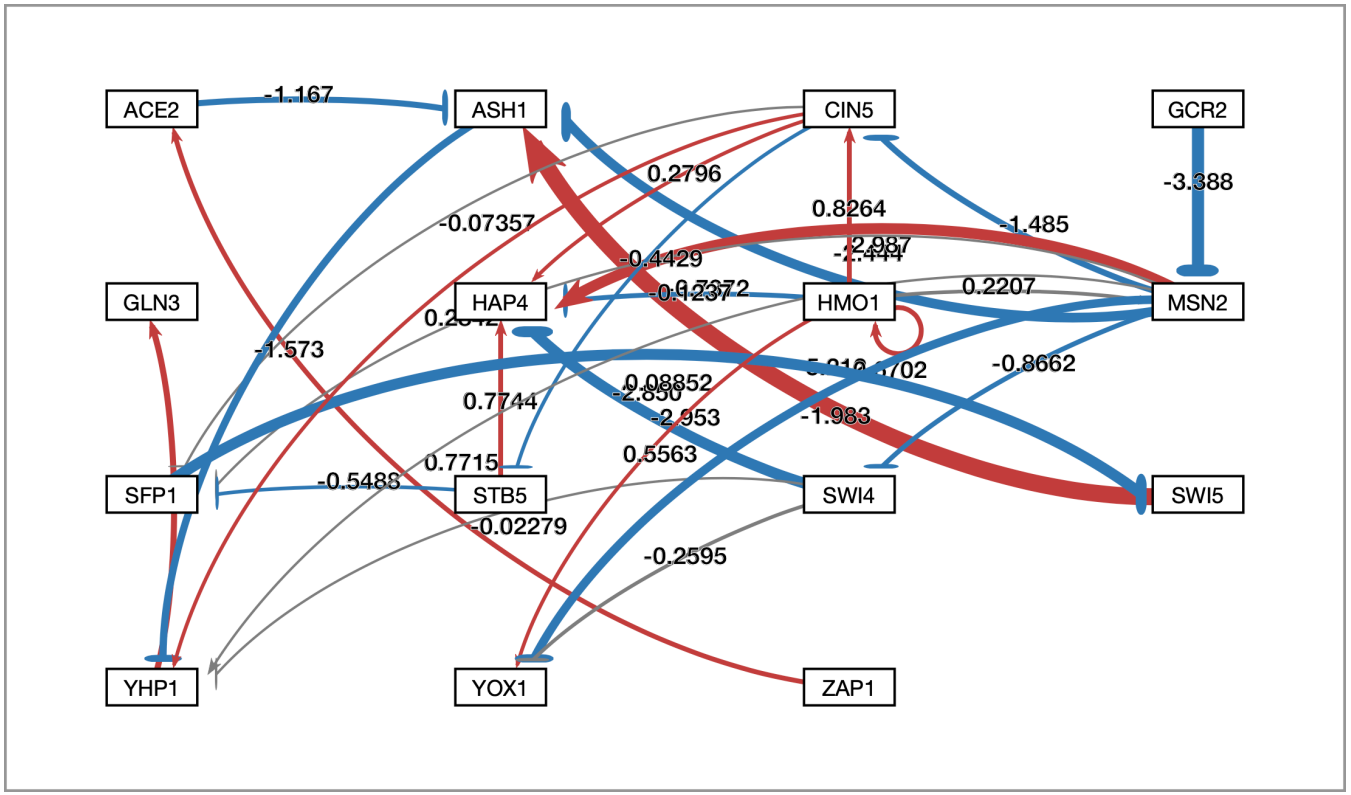
GRNsight now allows users to automatically lay out graphs in an easily readable grid format



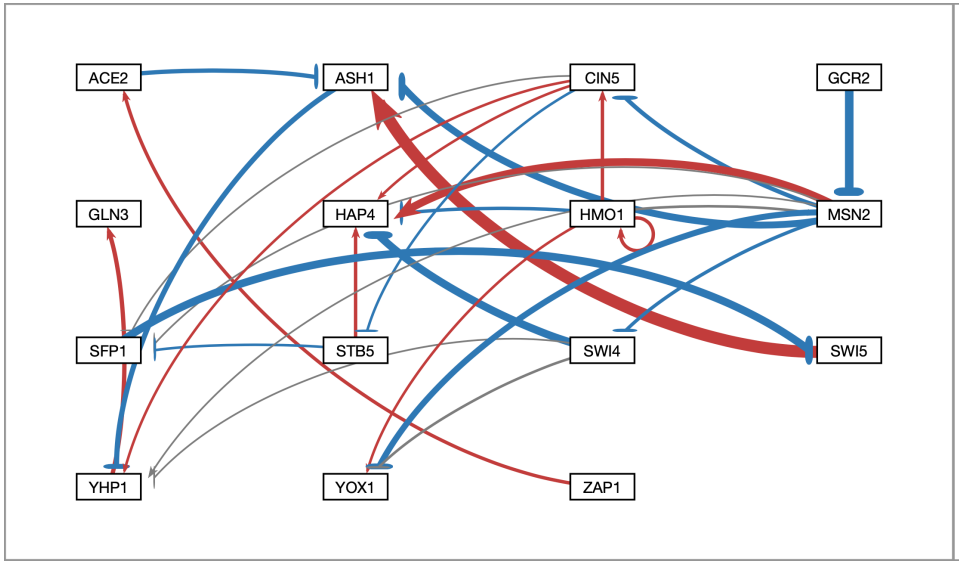
GRNsight now allows users to automatically lay out graphs in an easily readable grid format



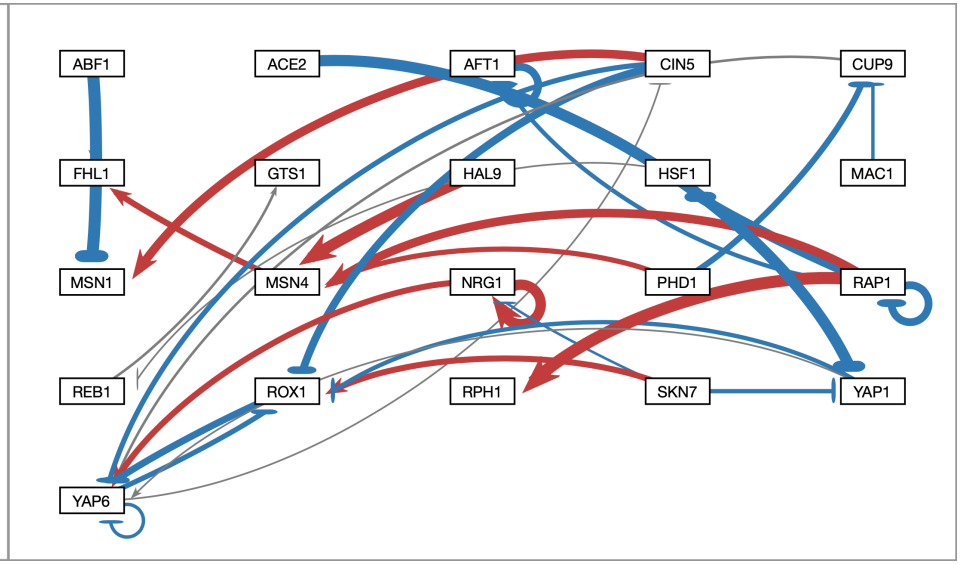
Numerical weight values can be displayed on the graph



Edge thicknesses are normalized to the largest magnitude weight value



Normalization Factor = 5.21

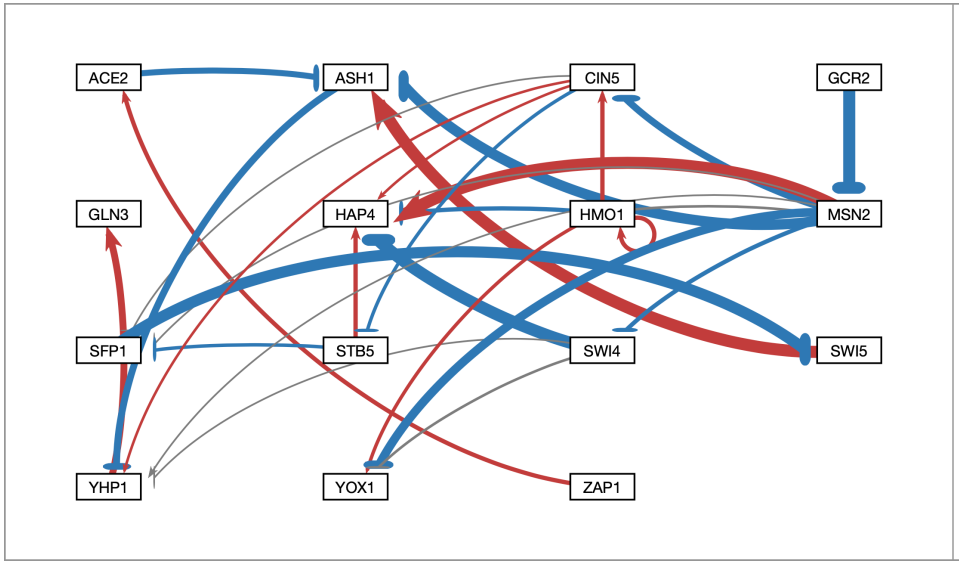


Normalization Factor = 1.5

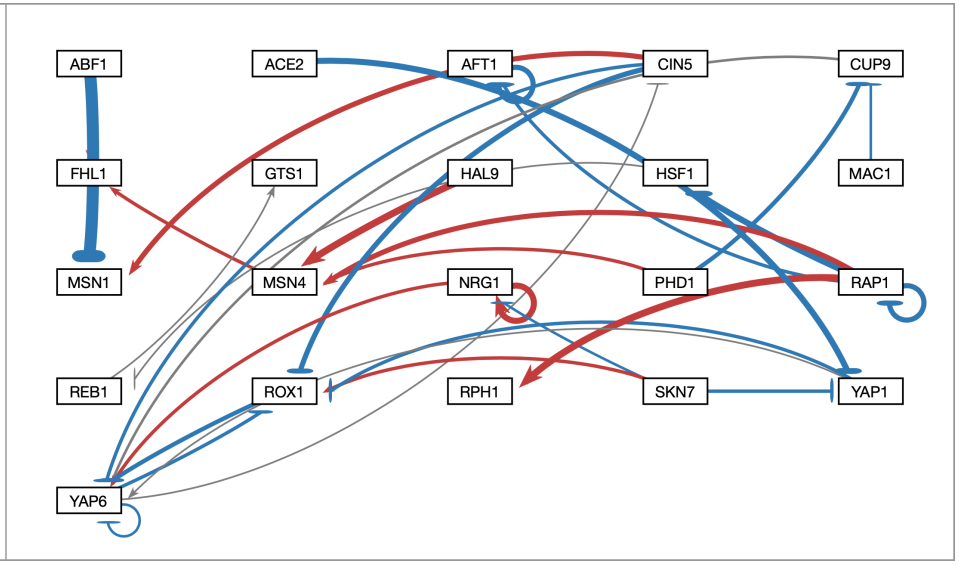
To facilitate better comparison of graphs, users may now modulate normalization factor

Edges are drawn relative to the highest magnitude edge weight, thus in order to facilitate accurate comparison between different graphs with different highest magnitude edge weight, the normalization factor may be changed.

Edge thicknesses are normalized to the largest magnitude weight value



Normalization Factor = 3

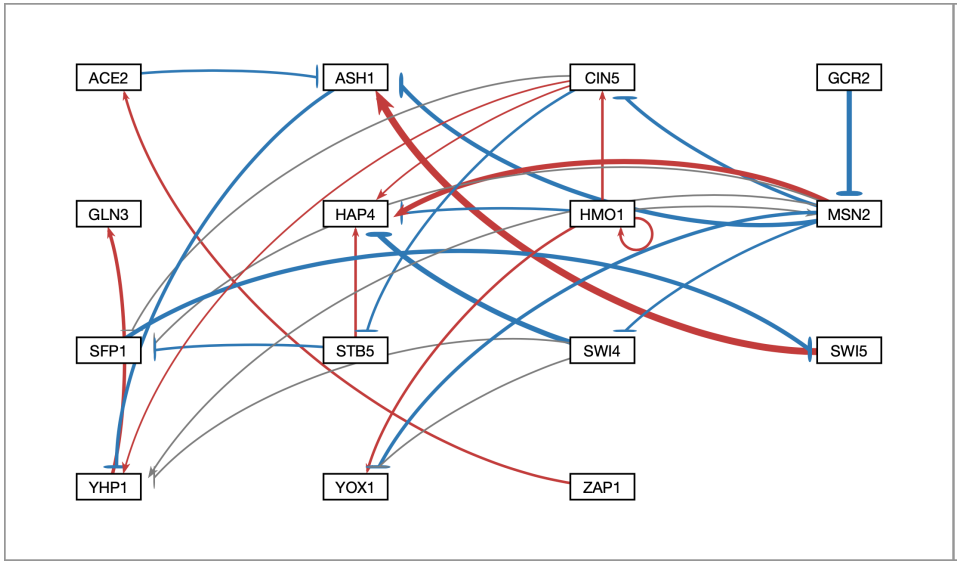


Normalization Factor = 3

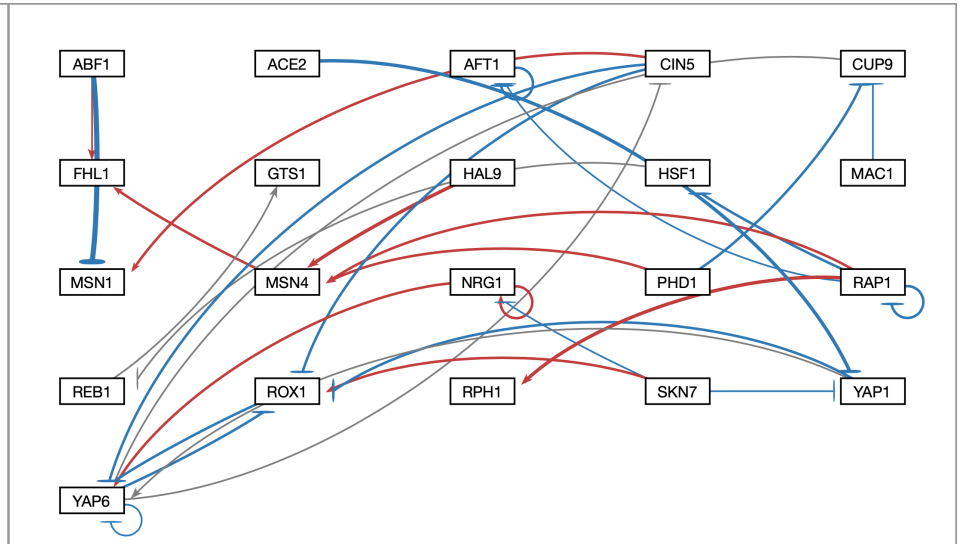
To facilitate better comparison of graphs, users may now modulate normalization factor

Edges are drawn relative to the highest magnitude edge weight, thus in order to facilitate accurate comparison between different graphs with different highest magnitude edge weight, the normalization factor may be changed.

Edge thicknesses are normalized to the largest magnitude weight value



Normalization Factor = 10

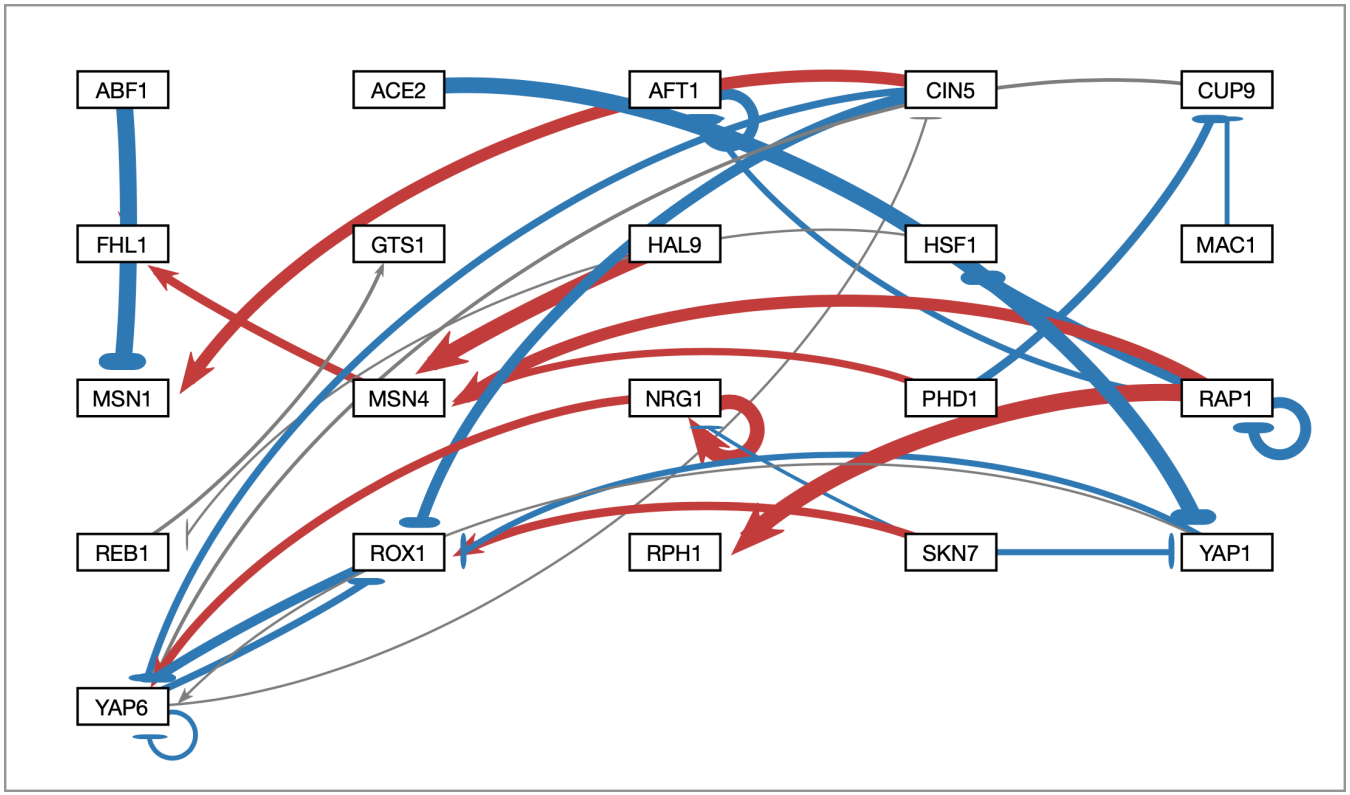


Normalization Factor = 10

To facilitate better comparison of graphs, users may now modulate normalization factor

Edges are drawn relative to the highest magnitude edge weight, thus in order to facilitate accurate comparison between different graphs with different highest magnitude edge weight, the normalization factor may be changed.

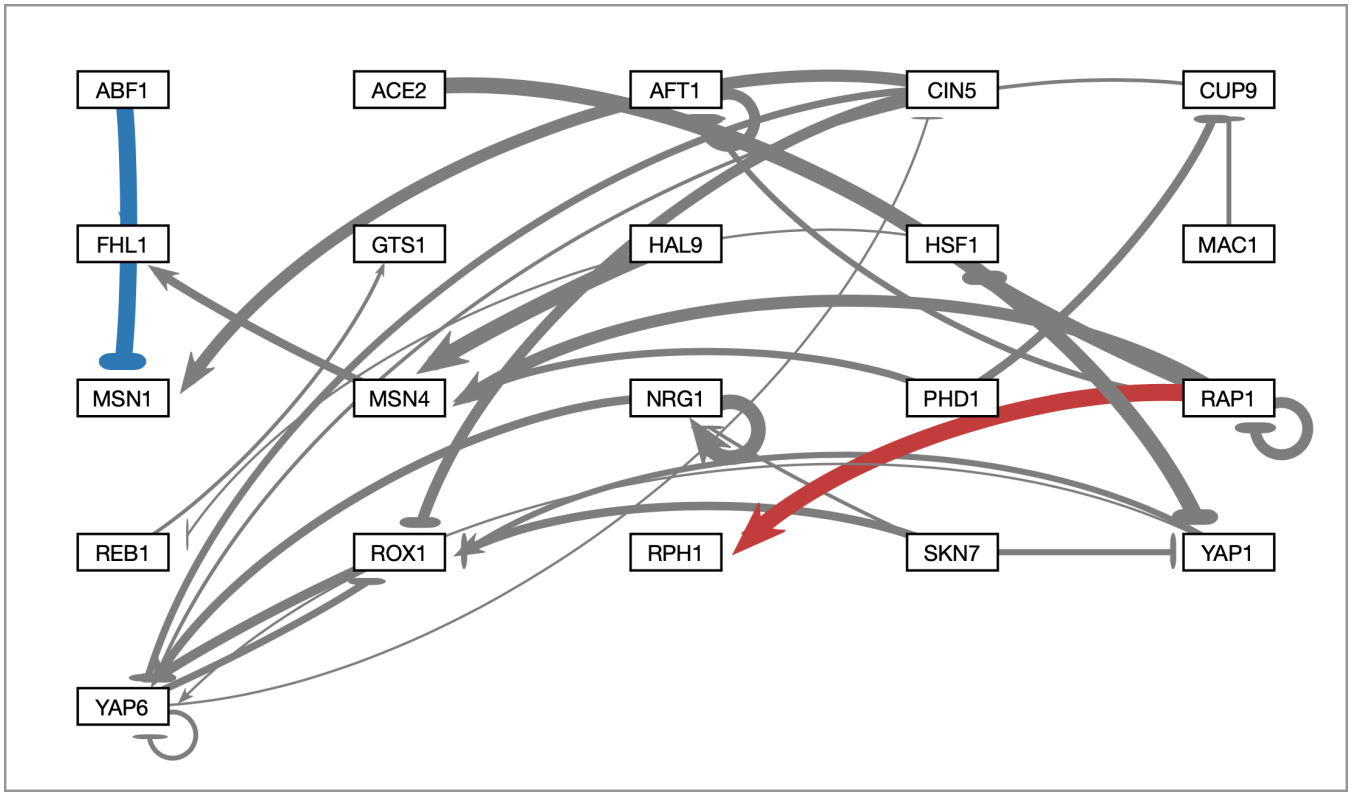
Edges with normalized weight values within 5% of 0 are displayed as gray to minimize their visual strength



To allow users to direct attention to certain nodes, users may now adjust gray edge threshold

Gray Edge Threshold = 5%

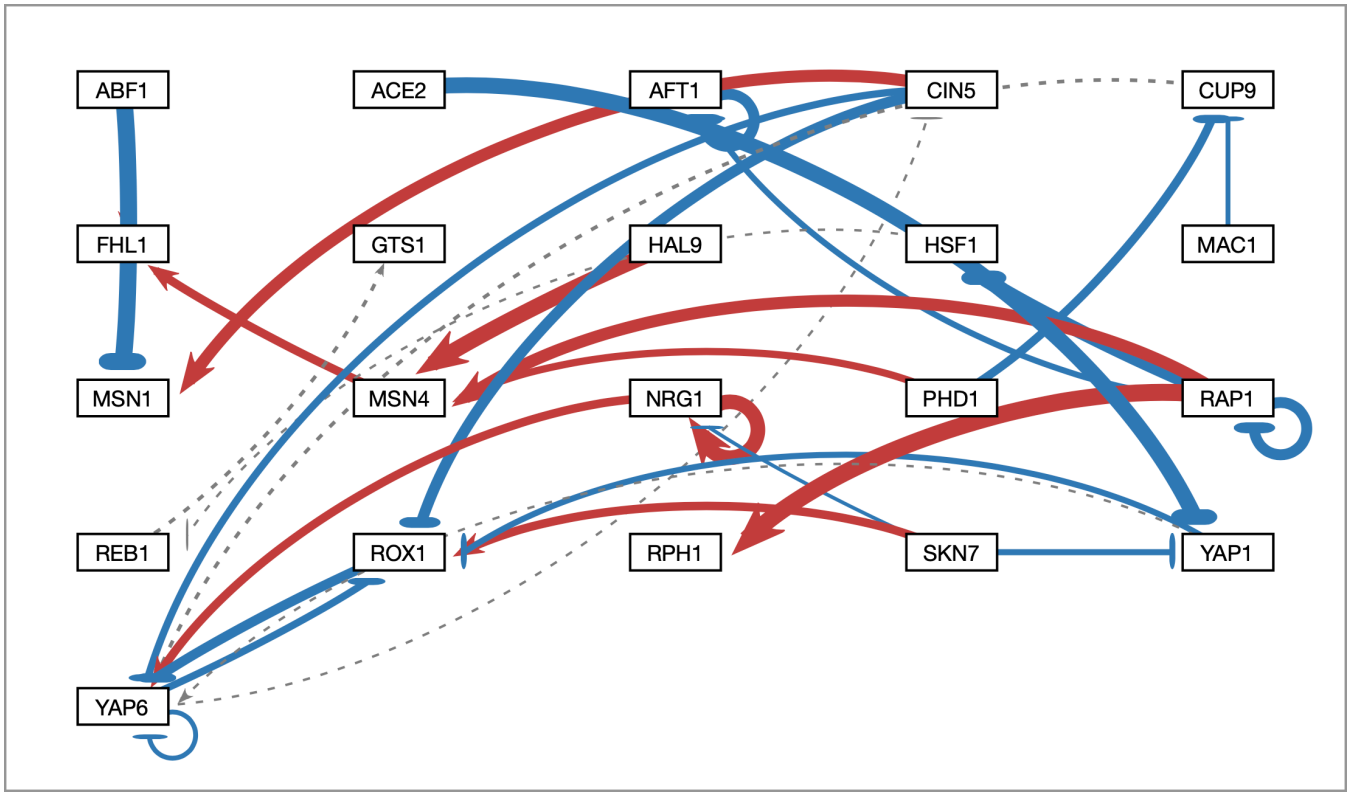
Users can change the gray edge threshold with a slider



To allow users to direct attention to certain nodes, users may now adjust gray edge threshold

Gray Edge Threshold = 50%

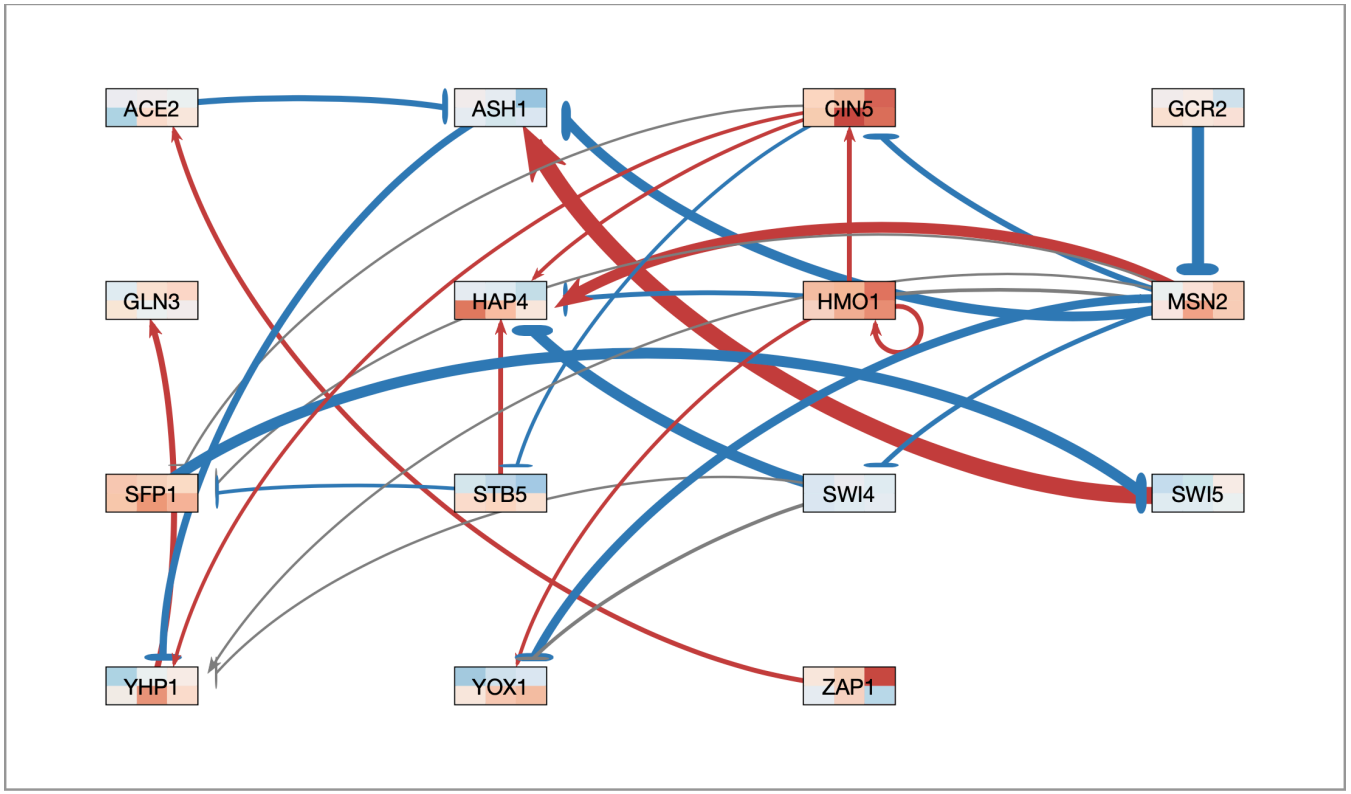
Gray edges can also be shown as dashed lines to facilitate visualization by color blind users



As a benefit to visually impaired users, gray edges may be displayed as dashed lines

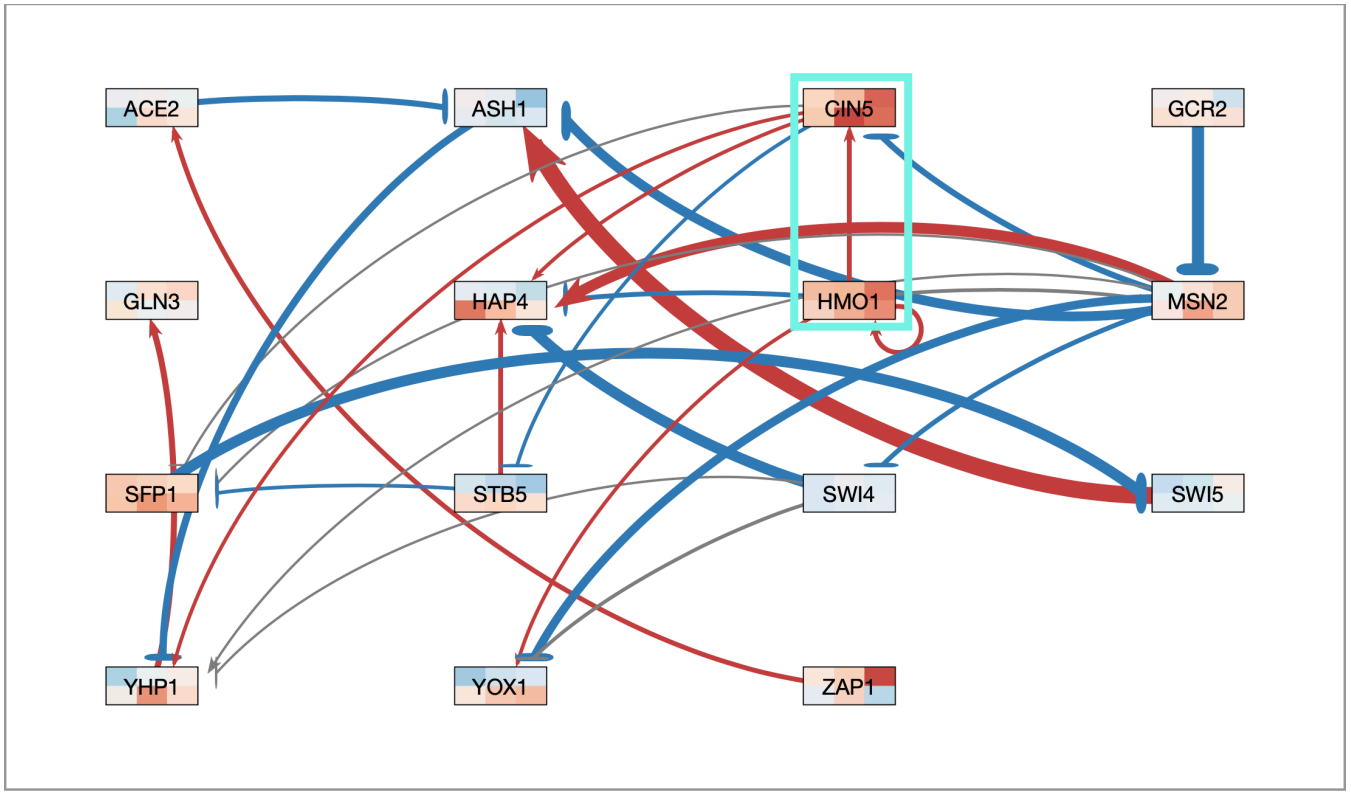
Gray Edge Threshold = 5%

Nodes now display measured and/or simulated expression data as individual heat maps



Top = wild type microarray data
Bottom = *Δ gln3* microarray data

Nodes now display measured and/or simulated expression data as individual heat maps



Top = wild type microarray data
Bottom = Δ *gln3* microarray data

Right-clicking on a node opens a web page which displays information retrieved from various databases

GLN3

Saccharomyces cerevisiae (strain ATCC 204508 / S288c)

General Information | Protein Information | Regulation | Gene Ontology | Sources

General Information

SGD ID:	S000000842 [1]
NCBI Gene ID:	856763[4]
Ensembl ID:	Not found[3]
Uniprot ID:	GLN3_YEAST[2]
JASPAR ID:	MA0307.1[5]
Description:	Transcriptional activator of genes regulated by nitrogen catabolite repression; localization and activity regulated by quality of nitrogen source and Ure2p [1]
Species:	Saccharomyces cerevisiae (strain ATCC 204508 / S288c)[2]
Locus Tag:	YER040W[4]
JASPAR Family:	GATA-type zinc fingers[5]
JASPAR Class:	Other C4 zinc finger-type factors[5]
Chromosome Sequence:	NC_001137.3[4]

Protein Information

Regulation

Gene Ontology

Sources

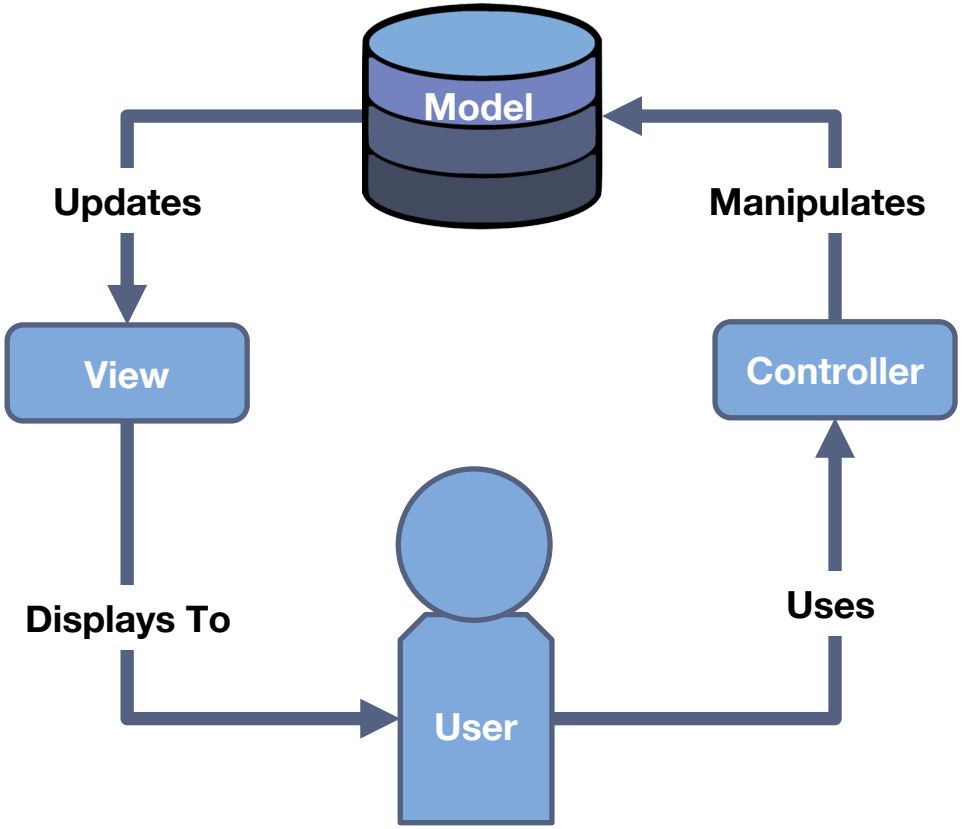
However…

In the past 5 years:

- GRNsight's codebase has expanded nearly 10x
- 13 new features have been added
- 11 different developers have worked on GRNsight

Lead to application instability, user interaction bugs

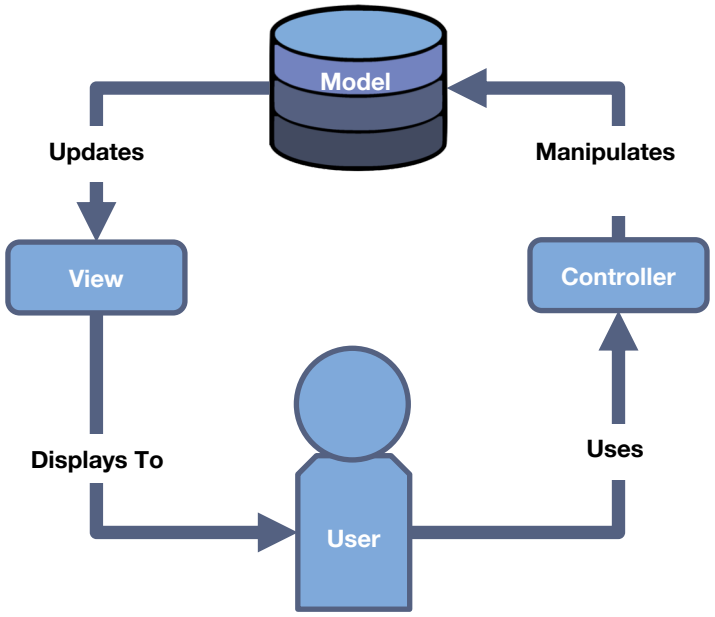
The Model-View-Controller paradigm is an architectural pattern used for developing user interfaces



The Model-View-Controller paradigm is an architectural pattern used for developing user interfaces

Creates separation of internal representations of information from the ways information is presented to and accepted from the user

Decouples the various components that comprise a complex application like GRNsight



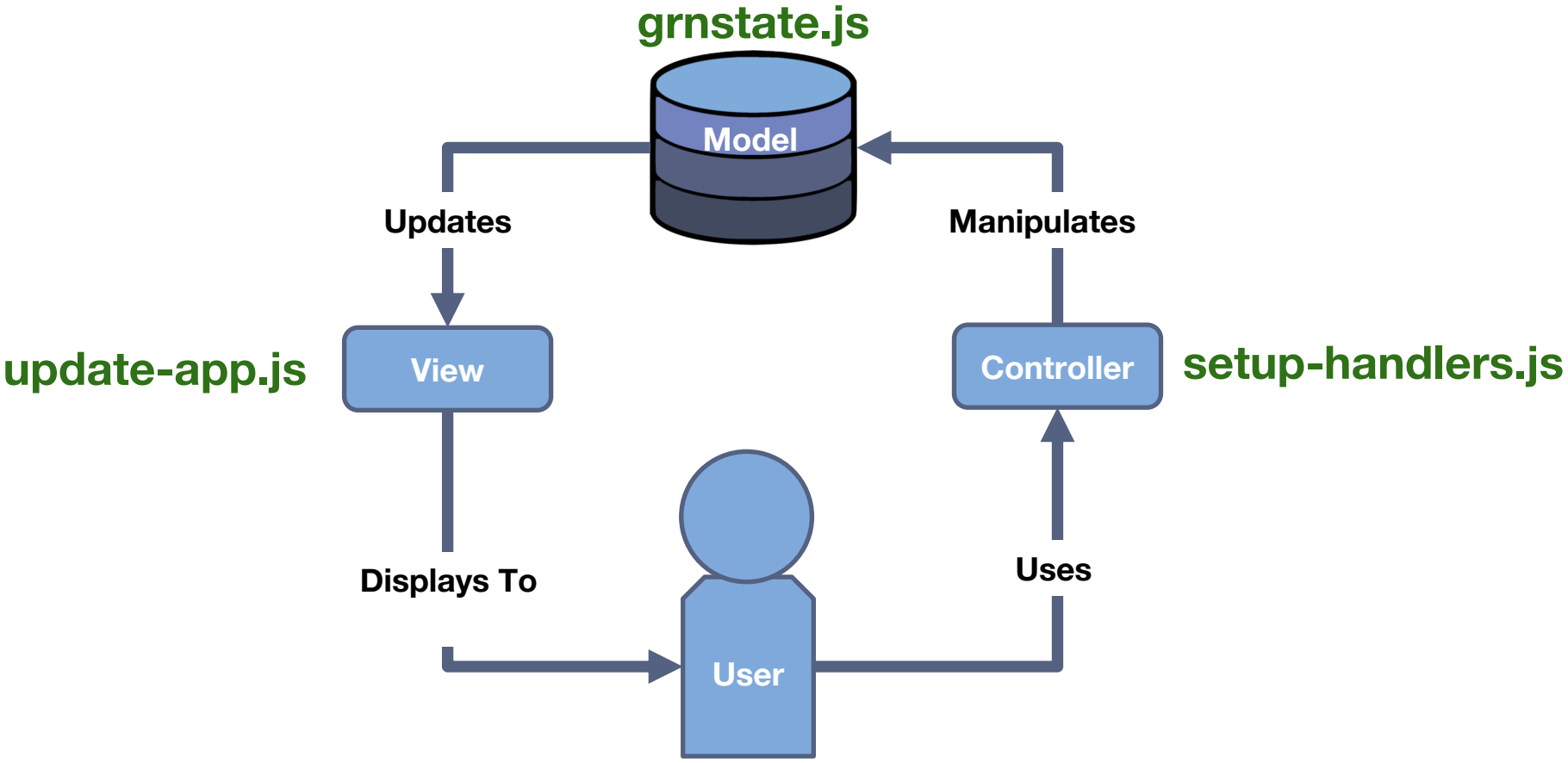
Why the MVC architecture?

Central store for application state & increased modularity:

- Reducing application state conflicts
- Diagnosing a bug
- Adding a new feature

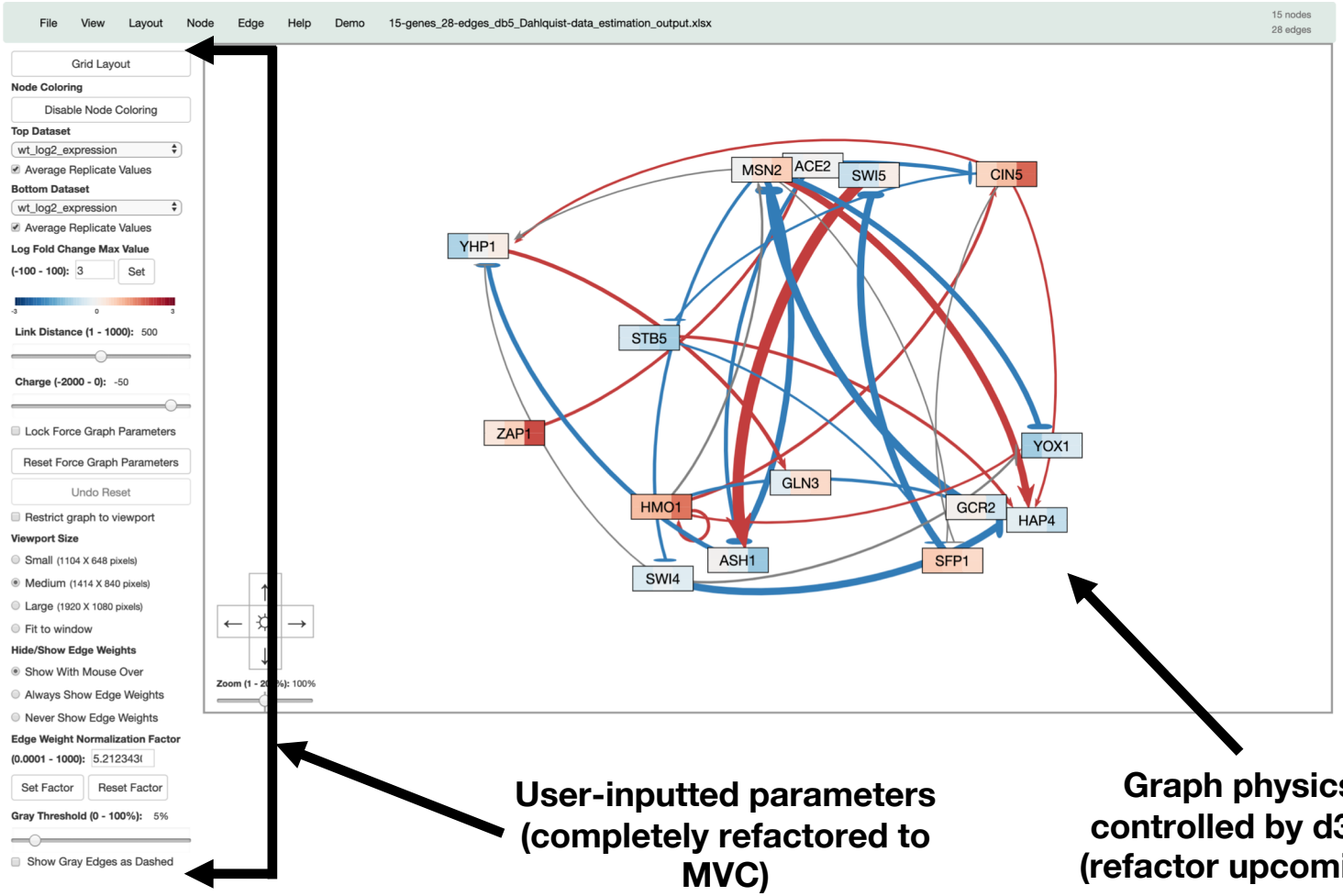
```
export const grnState = {
    name: null,
    simulation: undefined,
    newNetwork: false,

    get network () {
        return currentNetwork;
    },
    set network (network) {
        currentNetwork = network;
        this.resetNormalizationMax =
            max(network.positiveWeights.concat(network.negativeWeights));
        this.newNetwork = true;
    },
// Edge Display Parameters
    normalizationMax: null,
    resetNormalizationMax: null,
    edgeWeightDisplayOption: SHOW_WEIGHTS_MOUSEOVER,
    colorOptimal: true,
    grayEdgeThreshold: 5,
    dashedLine: false,
    annotateLinks: () => annotateLinks(currentNetwork),
// Node Coloring
    nodeColoring: {
        nodeColoringEnabled: true,
        logFoldChangeMaxValue: DEFAULT_MAX_LOG_FOLD_CHANGE,
        logFoldChangeUpdateTriggered: false,
        averageTopDataset: true,
        averageBottomDataset: true,
        topDataset: undefined,
        bottomDataset: undefined,
        lastDataset: null,
        bottomDataSameAsTop: true,
    },
// Slider Parameters
    slidersLocked: false,
    resetTrigger: false,
    undoResetTriggered: false,
    linkDistanceSlider: {
        sliderId: LINK_DIST_SLIDER_ID,
        valueId: LINK_DIST_VALUE,
        defaultVal: LINK_DIST_DEFAULT_VALUE,
        currentVal: LINK_DIST_DEFAULT_VALUE,
        backup: LINK_DIST_DEFAULT_VALUE,
        needsAppendedZeros: false,
        forceChanged: false,
    },
    chargeSlider: {
        sliderId: CHARGE_SLIDER_ID,
        valueId: CHARGE_VALUE,
        defaultVal: CHARGE_DEFAULT_VALUE,
        currentVal: CHARGE_DEFAULT_VALUE,
        backup: CHARGE_DEFAULT_VALUE,
        needsAppendedZeros: false,
        forceChanged: false,
    },
// Graph Layout Parameter
    graphLayout: "FORCE_GRAPH",
};
```

GRNsight's new central architectural pattern



12 files → 6 files

90% of the GRNsight codebase now follows a strict MVC architecture

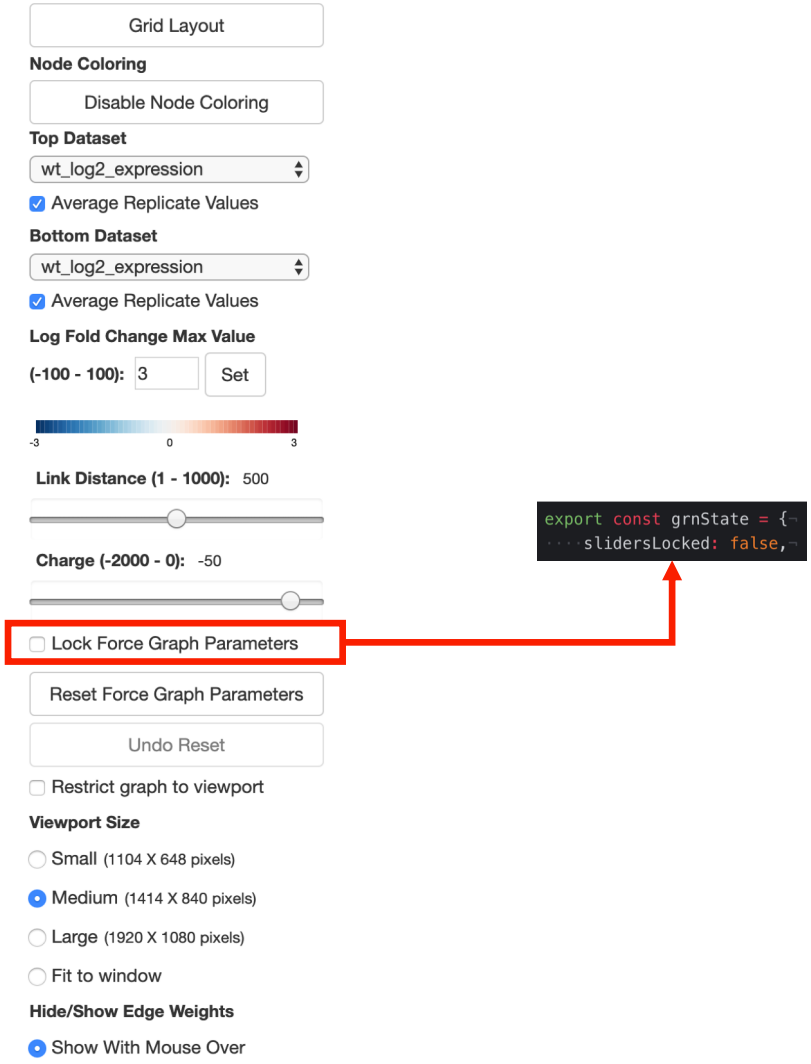


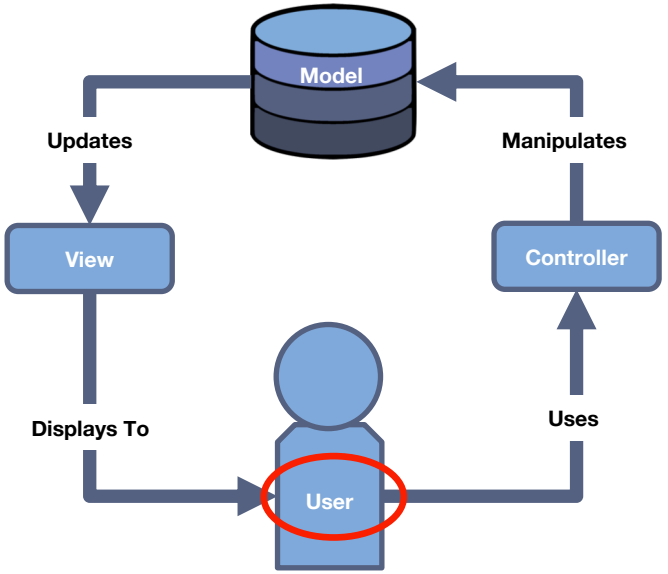
User-inputted parameters (completely refactored to MVC)

Graph physics controlled by d3.js (refactor upcoming)

The MVC refactor is “under the hood,” and is seamless to the user

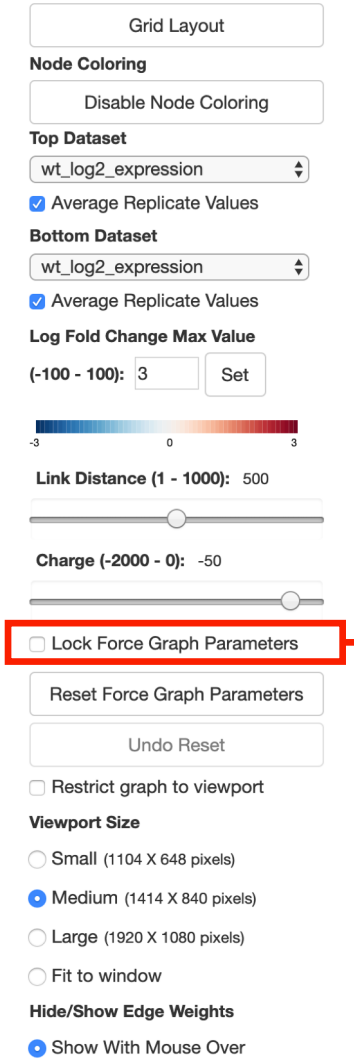
Locking Force Graph Parameter Sliders





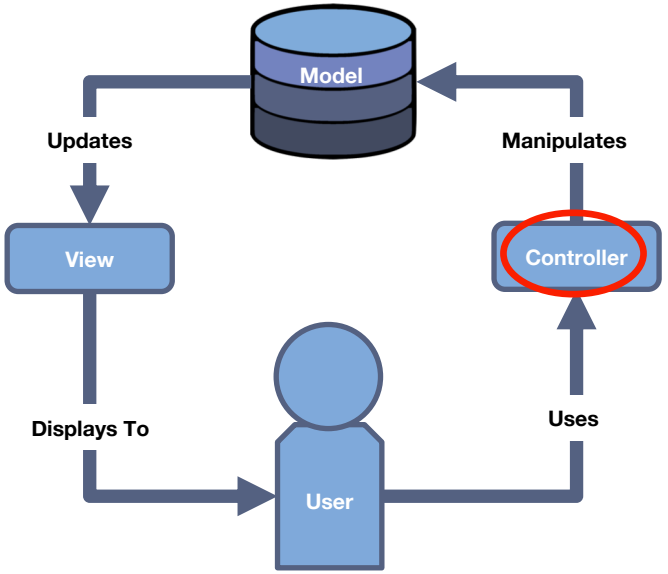
The MVC refactor is “under the hood,” and is seamless to the user

Locking Force Graph Parameter Sliders



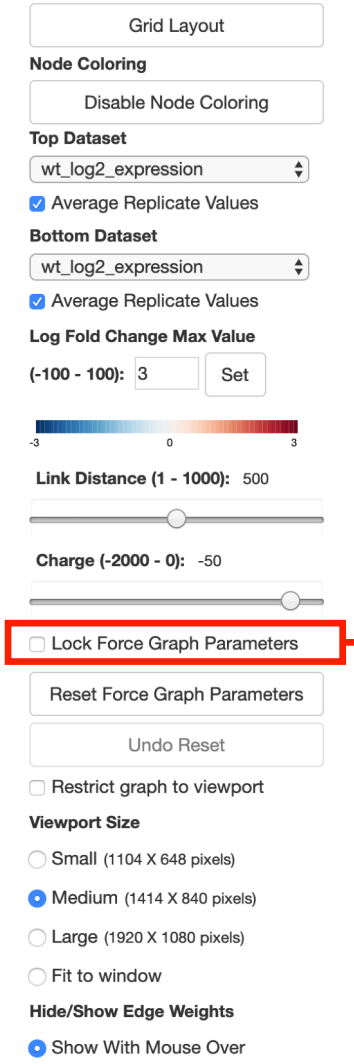

```
$(LOCK_SLIDERS_BUTTON).click(() => {
    grnState.slidersLocked = !grnState.slidersLocked;
    updateApp(grnState);
});
```

```
export const grnState = {
    slidersLocked: false,
```



The MVC refactor is “under the hood,” and is seamless to the user

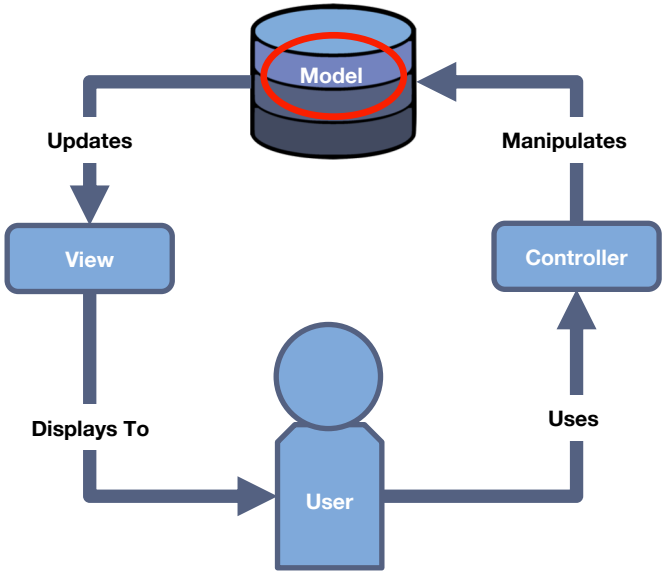
Locking Force Graph Parameter Sliders




```
export const grnState = {
    slidersLocked: true,
```

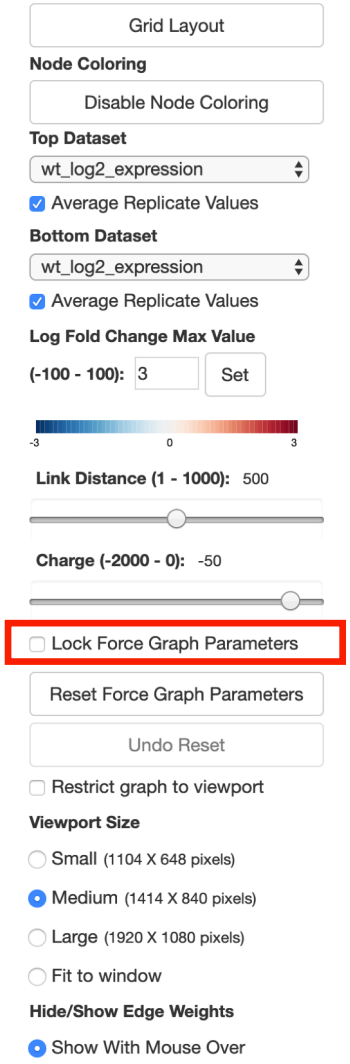
```
$(LOCK_SLIDERS_BUTTON).click(() => {
    grnState.slidersLocked = !grnState.slidersLocked;
    updateApp(grnState);
});
```

```
export const grnState = {
    slidersLocked: false,
```



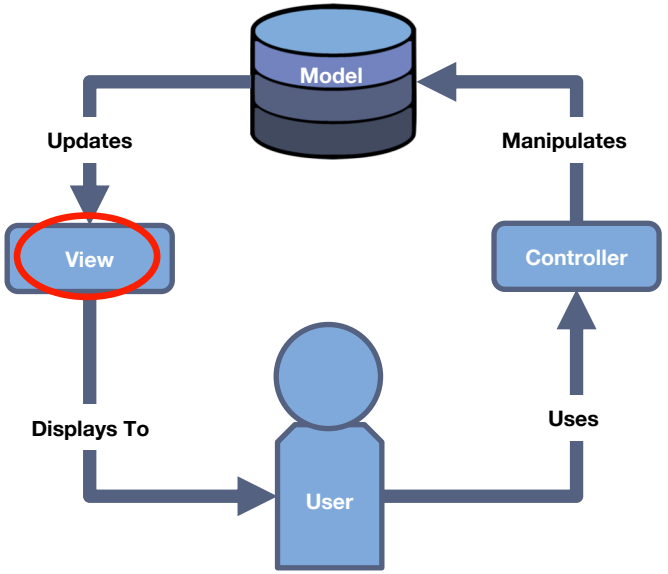
The MVC refactor is “under the hood,” and is seamless to the user

Locking Force Graph Parameter Sliders



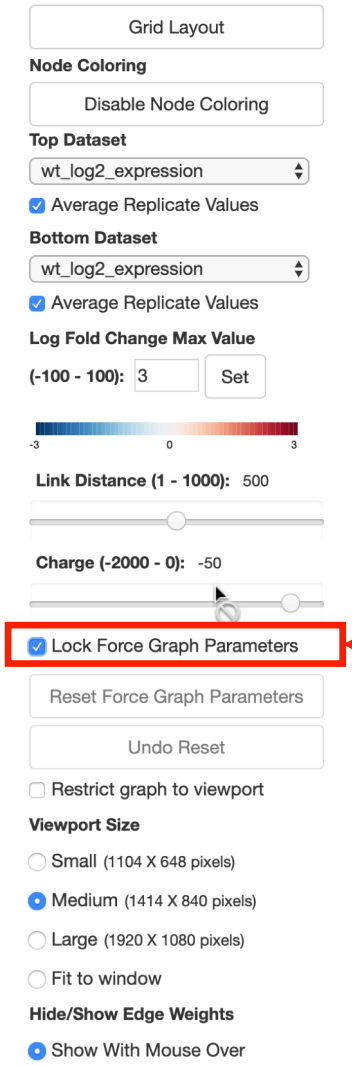

```
export const grnState = {
    slidersLocked: true,
```

```
if (grnState.slidersLocked === true) {
    $(LOCK_SLIDERS_MENU_OPTION + " span").removeClass("invisible");
    $(LOCK_SLIDERS_MENU_OPTION + " span").addClass("glyphicon-ok");
    $(RESET_SLIDERS_MENU_OPTION).parent().addClass("disabled");
    $(LINK_DIST_CLASS).parent().addClass("disabled");
    $(CHARGE_CLASS).parent().addClass("disabled");
    lockForce(grnState.slidersLocked);
} else {
    $(LOCK_SLIDERS_MENU_OPTION + " span").removeClass("glyphicon-ok");
    $(LOCK_SLIDERS_MENU_OPTION + " span").addClass("invisible");
    $(RESET_SLIDERS_MENU_OPTION).parent().removeClass("disabled");
    $(LINK_DIST_CLASS).parent().removeClass("disabled");
    $(CHARGE_CLASS).parent().removeClass("disabled");
    lockForce(grnState.slidersLocked);
}
const lockForce = (disable) => {
    $(LINK_DIST_SLIDER_ID).prop("disabled", disable);
    $(CHARGE_SLIDER_ID).prop("disabled", disable);
    $(RESET_SLIDERS_BUTTON).prop("disabled", disable);
    $(LOCK_SLIDERS_BUTTON).prop("checked", disable);
};
```



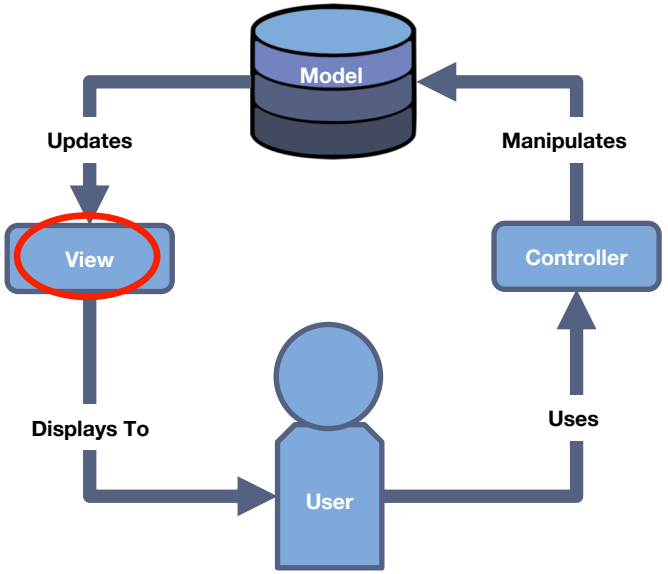
The MVC refactor is “under the hood,” and is seamless to the user

Locking Force Graph Parameter Sliders




```
export const grnState = {
    slidersLocked: true,
}
```

```
if (grnState.slidersLocked === true) {
    $(LOCK_SLIDERS_MENU_OPTION + " span").removeClass("invisible");
    $(LOCK_SLIDERS_MENU_OPTION + " span").addClass("glyphicon-ok");
    $(RESET_SLIDERS_MENU_OPTION).parent().addClass("disabled");
    $(LINK_DIST_CLASS).parent().addClass("disabled");
    $(CHARGE_CLASS).parent().addClass("disabled");
    lockForce(grnState.slidersLocked);
} else {
    $(LOCK_SLIDERS_MENU_OPTION + " span").removeClass("glyphicon-ok");
    $(LOCK_SLIDERS_MENU_OPTION + " span").addClass("invisible");
    $(RESET_SLIDERS_MENU_OPTION).parent().removeClass("disabled");
    $(LINK_DIST_CLASS).parent().removeClass("disabled");
    $(CHARGE_CLASS).parent().removeClass("disabled");
    lockForce(grnState.slidersLocked);
}
const lockForce = (disable) => {
    $(LINK_DIST_SLIDER_ID).prop("disabled", disable);
    $(CHARGE_SLIDER_ID).prop("disabled", disable);
    $(RESET_SLIDERS_BUTTON).prop("disabled", disable);
    $(LOCK_SLIDERS_BUTTON).prop("checked", disable);
};
```



Conclusion

- **GRNsight is an open source web application and service for visualizing models of gene regulatory networks.**
- **GRNsight has matured since version 1.0, adding 13 new features, and expanding its codebase by over 10 times.**
- **This necessitated refactor of 90% of GRNsight's codebase to the Model-View-Controller paradigm, leading to significantly smoother developer and user interactions**

GRNsight application and code are available under the open source BSD license

http://dondi.github.io/GRNsight/

https://github.com/dondi/GRNsight

Current and future work documented via Github Issues

- Move graph.js to MVC
- Import networks from gene databases
- Automate the creation of GRNmap workbooks

Acknowledgements

Dr. Kam D. Dahlquist
Dr. John David N. Dionisio

GRNsight Team
Alexia Filler
John Lopez
Justin Kyle Torres

GRNmap Team
Lauren Kelly

Wet Lab Team
Genesis Cruz