## SimChoice: a simulation module to help decide which assays will lead to the best network inference

Jesse et al.

## Introduction

 Gene regulatory networks can be inferred by observing how genes change expression over time and due to various perturbations. When one gene is up or down regulated, the genes that it affects may see changes in expression. A gene regulatory network can be inferred by observing changes in many genes over many experiments of different types. A network is of interest because of the understanding it affords and as a tool to predict how an organism will behave in untested conditions.

 There are three basic experiments one can perform: (i) genetic modifications of one or more genes, either constituitively or transiently; (ii) steady state experiments in which one changes the environment and then tests the expression response after the organism has reached steady state; and (iii) time series experiments which may result from any of the first two perturbation but in which expression assays are taken at several time points.

 Our tool combines information from the three experimental types using leading algorithmic methods with the possibility of adding other algorithms. The goal of the tool is to help researchers decide how many experiments and how many replicates of each experiment of the above three types to perform. We present her a case study of using this tool on the simulator for DREAM [ref].

Our SimChoice toolkit processes data in two ways. The first is to create “pipelines” of algorithms where the output network of one algorithm is used as a prior in another. The second approach is to run several different algorithms and use their output networks to “vote” on edges. The idea is that the strongest edges will show up across the output of different algorithms, reducing the noise in the inference.

## Data

 Gene expression data comes in many different types. The two coarsest types we will label “steady-state” and “time-series” data. A steady-state experiment involves measurement of gene expression at only a single time point, after the network has reached a “steady-state”. A steady-state is when the expression values of the genes have stopped changing and have reached an equilibrium.

 Time-series data are when a gene network is perturbed, as in steady-state, but gene expression values are recorded at a number of time points before reaching steady-state. This type of data is more difficult and expensive to collect, but allows researchers to model the dynamics of how the network is behaving over time.

 Within each of these coarse data types, the type of perturbation can differ. A gene can be removed (knocked out) from the network, or it forced to be over- or underexpressed (overexpression or knock down, respectively). Chemical perturbations, e.g., adding nitrogen to a plant, can also be used with no genetic manipulation.

## Algorithms in Current Version of SimChoice

### Steady-state algorithms

 The general approach for inferring gene regulatory networks form steady-state data is to look for correlations between a gene’s behavior and the behavior of other genes. For example, if five experiments are run and we see that when gene A has a high expression value, gene B has a low expression value, we may want to conclude that there is some sort of regulatory edge between A and B. From this example alone we cannot conclude that A’s high expression value is causing B’s low expression, or vice versa, but there may be a relation between the two. Some methods have ways of inferring directionality of edges, but many do not.

1 to 3 sentence blurbs about each of these

GENIE

MCZ

Etc.

### Time-series algorithms

 Time-series algorithms are a relative newcomer to the field. The approach here is to observe how the genes are changing over time and build a network that way. This allows a more fine-grained approach than with steady-state data: we can observe what happens at time T and build a model that predicts what will happen at time T+1. By doing this iteratively over the time points, a regulatory network can be constructed from the model weights.

DFG

Inferelator

### Pipelines

Inferelator Pipeline

**Methods: Case Study**

**Description of Networks to Be Inferred, variations of size, noise**

**Experimental Strategies considered**

**Results to see which Strategies performed best in terms of network edges and separately in terms of predicting out of sample data.**

## Methods

## Results