Weekly schedule: Math 9850, Fall 2024
Algebraic Systems Biology

- **WEEK 1: 8/21–8/23**.

  - **Wed. August 21**: We gave a brief history of mathematical biology, and asked what algebra (linear and nonlinear) has to do with biology? We saw a simple example of a biochemical reaction network, and how it define a system of nonlinear ODEs under the assumption of mass-action kinetics. We read over Michael Reed's 2015 article on *Mathematical biology is good for mathematics*, and covered the *What is algebraic biology?*" slides, pp. 1–14.

    Suggested reading: M.C. Reed. Mathematical biology is good for mathematics. *Notices of the AMS*, **62**(10): 1172–1176, 2015.

  - **Fri. August 23**: We covered the second and third algebraic biology topics. First, we discussed the emergence of algebraic statistics since the turn of the century. This is especially useful for problems involving discrete probability, with genomics and phylogenetics as examples. We also mentioned the newer fields of topological data analysis, and what persistence homology is all about.

    Next, the third of three topics is algebraic neuroscience. Neural binary codes are defined by regions of place fields, and a challenging problem is to reconstruct the regions just from the code. These regions can be described combinatorially, geometrically, and algebraically. The latter is done with so-called neural ideals, which are generated by pseudomonomials. This has opened up a whole new area of mathematics in the past decade, which likely would not have been considered without the biological application.

    Finally, we just started the topic of modeling gene regulatory networks. We introduced the lactose (*lac*) operon in *E. coli*, which controls the genes for the transport and metabolism of lactose sugar in the cell. We described the biological processes, how these can be written as biochemical reactions, and the resulting system of differential equations. In a few weeks, we'll explore this in a lot more detail. We covered the *What is algebraic biology?*" slides, pp. 15–27.

    Suggested reading: M. Macauley. A case for algebraic biology: from research to education. *Bull. Math. Biol.*, **82**(115), 1829–1852, 2020.

- **WEEK 2: 8/26–8/30**.

  - **Mon. August 26**: We continued our preview / overview with models of the *lac* operon. Starting with a published 5-variable ODE models, we discussed

the concept of bistability, and how this model exhibits it. Next, we showed how Boolean models are an alternative framework. As a teaser, we saw some how computational algebra can be used to solve reverse engineering problems, such as the popular network inference problem in systems biology.

Suggested reading: R. Laubenbacher and B. Sturmfels. Computer algebra in systems biology. *Amer. Math. Monthly* **116**(10): 882–891, 2009.

– **Wed. August 28**: We spent most of the day with class introductions, before moving into chemical reaction networks. We discussed the law of mass-action kinetics, and showed how to derive a system ODE from checmical reaction network under this assumption. We covered the *Chemical reaction networks*" slides, pp. 1–6.

Suggested reading: Sections 1–2 of A. Dickenstein. Biochemical reaction networks: an invitation for algebraic geometers. Mathematical Congress of the Americas. *Contemp. Math.* 656: 65-83, 2015.

– **Fri. August 30**: We saw the classic Michaelis-Menten equation for an enzyme-substrate chemical reaction, and derived and analyzed the solution. Next, we considered the "higher order" case when the enzyme-substrate complex is ES_n rather than just ES. The solutions are Hill functions, and we discussed several variants, such as when a single enzyme catalyzes multiple reactions, or when the enzyme-substrate complex ES_2 forms in two sequential steps. We covered the *Chemical reaction networks*" slides, pp. 7–17.

Suggested reading: Sections 3–5 of A. Dickenstein. Biochemical reaction networks: an invitation for algebraic geometers. Mathematical Congress of the Americas. *Contemp. Math.* 656: 65-83, 2015.

• **WEEK 3: 9/2–9/6**.

– **Mon. September 2**: Labor day (no class)

– **Wed. September 4**: We reviewed the central dogma from molecular biology—the 2-step process of transcription (mRNA synthesis) and translation (protein synthesis). Next, we talked about operons, which are clusters of genes that are co-transcribed, and are primarly found in prokaryotes. We reviewed the *lac* operon in *E. coli*, which controls the genes responsible for the transport and metabolism of lactose. This is an *inducible* operon because it is off by default. In contrast, the *trp* operon is *repressible*, because it is on by default. We discussed the *lac* operon in more details than before, and included the *catabolite repression* mechanism, which inactives the promotor when glucose,

the prefered sugar, is present. We covered the entirity of the *Gene regulation by operons*" slides, pp. 1–15.

Suggested reading: Robeva/Hodge (2013), Chapter 1: *Mechanisms of Gene Regulation: Boolean Network Models of the Lactose Operon in* Escherichia coli, by R. Robevea. Sections 1.1–1.2, pp. 1–6.

– **Fri. September 6**: We discussed bistability and how it is exhibited in the *lac* operon, under medium levels of lactose concentration. Next, we began developing a model of the *lac* operon using delay differential equations (DDEs). These are like ODEs, but with time delays, e.g., $y'(t) = ky(t - t_0)$, rather than $y'(t) = ky(t)$. They arise because biological processes like transcription and translation, are not instantaneous. First, we discussed how our model will account for decrease in concentration due to dilution (from cellular growth), and degredation of the gene products. Next, we derived two ODEs from chemical reaction networks, of (1) the repressor binding to allolactose, and (2) the repressor binding to the operator region. We derived a formula for $[O]/O_{tot} = \frac{1}{1+K_2[R]}$, the proportion of free operator sites, showing that it cannnot be absolutely zero. We covered the *Delay differential equation models of gene regulation*" slides, pp. 1–9.

Suggested reading: Robeva/Hodge (2013), Chapter 2: *Bistability in Lactose Operon of* Escherichia coli*: A Comparison of Differential Equation and Boolean Network Models*, by R. Robeva and N. Yildirim. Sections 2.1–2.3, pp. 37–46.

• **WEEK 4: 9/9–9/13**.

– **Mon. September 9**: We finished modeling the *lac* operon dynamics using mass-action kinetics, and got a Hill-like equation for the proportion of free operator sites, in terms of the concentration $[A]$ of the inducer, allolactose. Next, we saw how to model time-delays in a basic exponential growth model. Finally, we saw the 3-variable delay differential equation (DDE) model of the operon, by Yildirim and Mackey. The variables are M (mRNA), B (β-galactosidase), and A (allolactose). There is one parameter (constant), which is L (lactose). We derived the differential equation for each of these.

The first step with such a model is to find the fixed points, though these had to have been done numerically. For a middle range of lactose, there are three fixed points, which is what one would expect for bistability. For lower or higher ranges of lactose, there is only one fixed point. This is nicely illustrated on a 1-parameter bifurication diagram of A^* vs. L, as well as the simulated time series for $M(t)$, $A(t)$, and $B(t)$. We covered the *Delay differential equation*

models of gene regulation" slides, pp. 9–18.

Suggested reading: Robeva/Hodge (2013), Chapter 2: *Bistability in Lactose Operon of* Escherichia coli*: A Comparison of Differential Equation and Boolean Network Models*, by R. Robeva and N. Yildirim. Section 2.4, pp. 47–57.

– **Wed. September 11**: We introduced a 5-variable DDE model of the *lac* operon. This one uses M, B, and A as before, but also L (lactose) and P (*lac* permease). There is one parameter, L_e (extracellular lactose). The long-term results with the fixed points and bistability are analogous. We also briefly saw a few other DDE models of operons, such as the tryptophan (*trp*), tryptophanse (*tna*), and arabinose (*ara*). The authors of these papers come from math, physics, and engineering.

Next, we introduced the idea of a Boolean model, with a "toy model" of the *lac* operon, with three nodes and two parameters. The functions are updated synchronously to yield a dynamical system map $f\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$, and the *state space* is a graph with vertex set $\mathbb{F}_2^n$ and 2^n directed edges, each of the form $(x, f(x))$. We covered the *Delay differential equation models of gene regulation*" slides, pp. 19–29, and *Basics of Boolean modeling*, pp. 1–6.

Suggested reading: Robeva/Hodge (2013), Chapter 1: *Mechanisms of Gene Regulation: Boolean Network Models of the Lactose Operon in* Escherichia coli. Section 1.3.1–1.3.3, pp. 6–15.

– **Fri. September 13**: We showed how to generated the state space of our toy Boolean model with Cyclone. By making the parameters constants, there were 5 variables, and the state space has four attractors, which are all fixed points that we expected biologically. Next, we proposed a more refined model with five variables, but it had a flaw because there were two fixed points that di not make biological sense. We also showed to to use Bool Net in R to find the attractors and plot the state space space. We covered the slides, *Basics of Boolean modeling*, pp. 6–16.

Suggested reading: Robeva/Hodge (2013), Chapter 1: *Mechanisms of Gene Regulation: Boolean Network Models of the Lactose Operon in* Escherichia coli, by R. Robeva. Sections 1.3.4–1.3.7, pp. 16–25.

• **WEEK 5: 9/16–9/20**.

– **Mon. September 16**: First, we went over a list of published Boolean networks, including operons in *E. coli*, and in other model organisms, such as the fruit fly, *Arabidopsis thaliana* (plant), yeast, *C elegans* (worm). There was

even an example from ecology. We mentioned a few example of well-modeled biological processes, such as genetic switches, cell cycles, and cell differentiation.

Next, we browsed the website of the Gene Interaction Network simulation (GINsim) software, which has been around since around 2005–06. This simulates *logical models*, a larger class of discrete models that includes Boolean networks. GINsim is freely available and can be run on Windows, Mac, or Linux. Models are stored in `.zginml` files, and the website has a repository of over 65 published models, with links to the corresponding papers. We browsed a few of things such as the lysis-lysogeny switch in the phage lambda virus (Thieffry/Thomas, 1995, w/ 339 citations), the Th-cell differentiation network (Mendoza, 2006, w/ 235 citations), a Boolean model of geroconversion (Verlingue et al., 2016), and the p53-Mdm2 network for DNA repair (Abou-Jaoudé et al., 2009, w/ 130 citations). It was striking at how different these papers were stylistically, which should be expected because they were written by researchers in different fields.

We also discussed how there are a number of frameworks for stochastic Boolean models. One example are *probabilistic Boolean networks*, or PBNs (Dougherty / Shmulevich, 2010), that put a probability distribution on the update functions. In another, *Stochastic Discrete Dynamical Systems*, or SDDSs (Murrugarra / Aguilar, 2018), there are activiation and inhibition probability parameters. The Markovian Boolean Stochastic Simulator (MaBoSS) software package simulates a continuous-time Markov chain on a Boolean network (Stoll et al., 2012). Using Google, we stumbled upon stochastic Boolean networks SBNs (Liang / Han, 2012). We covered the slides, *Basics of Boolean modeling*, pp. 16–17.

Suggested reading: Browse the GINsim website at `http://ginsim.org/`, especially the model repository.

– **Wed. September 18**: We began with a 9-variable Boolean model of the *lac* operon that had two parameters: L_e for extracellular lactose and G_e for extracellular glucose. The fixed points are the solutions of the system $f_{x_1} = x_1, f_{x_9} = x_9\}$. We saw how to solve this with the computational algebra software Macaulay2: compute a Gröbner basis of the ideal $I = (f_{x_1} + x_1, \ldots, f_{x_9} + x_9)$. For all 4 Boolean parameter vectors (G_e, L_e), there was one fixed point that was exactly what shoudl be expected biologically. We covered the slides *Fixed points of Boolean models*, pp. 1–14.

Suggested reading: Robeva/Hodge (2013), Chapter 1: *Mechanisms of Gene Regulation: Boolean Network Models of the Lactose Operon in* Escherichia

coli, by R. Robeva. Sections 1.4–1.6, pp. 25–34.

– **Fri. September 20**: We gave a brief overview of what Gröbner bases can do, without actually defining them. For our purposes, we can think of them as a "nonlinear version" of Gaussian elimination, in that they can be used to reduce a system of nonlinear polynomials, into a much simpler "upper triangular form." We saw how to compute a Gröbner basis in Macaulay2, and use it to solve such a nonlinear system of equations.

Next, we saw a published model of the *lac* operon that has 10 variables and 3 parameters, and exhibits bistability because there are two fixed points under medium levels of lactose. This model incorporates medium levels by adding additional Boolean variables, and we discussed other approaches to this, like using ternary variables, or multivariate logic. We used Macaulay2 to compute the fixed points of this model, which make sense biologically. We covered the slides *Fixed points of Boolean models*, pp. 12–14, and *Advanced features of Boolean models*, pp. 1–7.

Suggested reading: B. Sturmfels. What is. . . a Gröbner basis?" *Notices Amer. Math. Soc.* **52**(10):1199:–1200, 2005. `https://math.berkeley.edu/~bernd/what-is.pdf`, and the Introduction (pp. 1–3) to the preprint: J.G. Galofre, M. Pérez-Millán, A.G. Rial, R. Laubenbacher, A. Dickenstein. Beyond Boolean networks. 2024. `https://arxiv.org/pdf/2404.16760`

• **WEEK 6: 9/23–9/27**.

– **Mon. September 23**: We saw how time delays can be incorporated into Boolean models, both for activators and repressors. There are several ways to do this, depending on whether the time-delay of activation (resp., repression) is the same as of un-activation (resp., un-repression). Then, we saw how to incorporate diluation and degregation into Boolean models. Next, we revisited the 3-variable ODE models of the *lac* operon, and computed the half-lives of the variable (concentrations) using a basic ODE, in preparation for building a Boolean models using the variables (M, B and A), the incorporate time delays and dilution/degregation. We covered the slides *Advanced features of Boolean models*, pp. 7–14.

Suggested reading: Robeva/Hodge (2013), Chapter 2: *Bistability in Lactose Operon of* Escherichia coli*: A Comparison of Differential Equation and Boolean Network Models*, by R. Robeva and N. Yildirim. Section 2.5, pp. 57–60.

– **Wed. September 25**: We proposed several Boolean models of the *lac* operon, based off of the 3- and 5-variable ODE models. These incorporated time delays

for activation and dilution/degregation. They also exhibited bistability. We covered the slides *Advanced features of Boolean models*, pp. 15–26.

Suggested reading: Robeva/Hodge (2013), Chapter 2: *Bistability in Lactose Operon of* Escherichia coli*: A Comparison of Differential Equation and Boolean Network Models*, by R. Robeva and N. Yildirim. Sections 2.6–2.7, pp. 60–73.

- **Fri. September 27**: Class canceled (Hurricane Helene).

- **WEEK 7: 9/30–10/4**.

  - **Mon. September 30**: Class canceled (Hurricane Helene).

  - **Wed. October 2**: Many Boolean models are prohibitively large to effictively analyize. We saw how models can be reduced while preserving key features of their dynamics. Our focus was on a method that preserves the number of fixed points, which can then be recovered by back-substituting. We saw how to use Macaulay2 to do this efficiently. We covered the slides *Reduction of Boolean models*, pp. 1–16.

    Suggested reading: Robeva (2015), Chapter 6: *Steady State Analysis of Boolean Models: A Dimension Reduction Approach*, by A. Veliz-Cuba and D. Murrugarra. pp. 121–38.

  - **Fri. October 4**: We reduced a 26-node Boolean model of the T-helper cell differentiation network to a 2-node network that had 3 fixed points, which made sense biologically.

    We began the section where we will be formalizing algebraic models and finite dynamical systems, and we started with seeing the (state) state space and asynchronous automaton of several examples. The fixed points are always independent of update order. However, sometimes larger limit cycles in the synchronous state space can disappear in the asynchronous state space. Conversely, we saw an example of a published model (Faure et al, 2006) that had 1 fixed point and a 7-cycle in the synchronous state space, but a 112-node complex attractor in the asynchronous automaton.

    We introduced the concept of a *field*, which is a set in which we can add, subtract, multiply, and divide (but not by zero), and the distributive law holds. More gerally, a *ring* is such a set, but without the ability to divide (i.e., there need not be multiplicative inverses), and multiplication need not commute. We covered the slides *Reduction of Boolean models*, pp. 17–21, and *Algebraic*

models and finite dynamics systems, pp. 1-13.

Suggested reading: Faure, A., Naldi, A., Chaouiya, C. and Thieffry, D. (2006) Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22, e124-e131.

- **WEEK 8: 10/7–10/11**.

  - **Mon. October 7**: We spent the day talking about finite fields. Specifically, every the *characteristic* of finite field is the smallest $n \cdot 1 = 1 + \cdots + 1 = 0$, and this must be prime. We stated, without proof, that there is a unique finite field of order $q = p^n$ for each prime p and positive integer n. We showed how to construct this finite field as a quotient a polynomial ring, namely $\mathbb{F}_q \cong \mathbb{Z}_p[x]/\langle f(x)\rangle$, where $f(x)$ is any irreducible polynomial of degree n. We did some examples to see what this really means, both by hand, and using the Macaulay2 software package. We covered the slides *Algebraic models and finite dynamics systems*, pp. 14-17 (mostly board work).

  - **Wed. October 9**: Fields like $\mathbb{Q}$ and $\mathbb{R}$ are *totally ordered*, whereas $\mathbb{C}$ is not. Though we often consider $0 < 1 < 2 < \cdots < p-1$ in $\mathbb{F}_p$, this is not an "actual" total order.

    We spent most of the class understanding the difference between the infinite ring of polynomials $\mathbb{F}[x]$ vs. the finite set of functions $\mathbb{F}_q \to \mathbb{F}_q$. We saw how to represent every such function with a *truth table*, and a basic counting argument shows that there are q^q such truth tables.

    Over the field $\mathbb{F}_q$, we always have $x^q = x$, and so any two polynomials that different by replacing x^q with x will define the same function. This means that every function $\mathbb{F}_q \to \mathbb{F}_q$ can be represented by a polynomial in the quotient ring $\mathbb{F}[x]/\langle x^q - x\rangle$. Note that both of these sets have size q^q. We covered the slides *Algebraic models and finite dynamics systems*, pp. 18-20 (mostly board work).

    Suggested reading: Robeva/Macauley (2018), Chapter 4: *The Regulation of Gene Expression by Operons and the Local Modeling Framework.* Sections 4.3.1.–4.3.2, pp. 106–112.

  - **Fri. October 11**: The representation between Boolean (or ternary, etc.) functions and elements in a quotient ring extends naturally to multivariate polynomials. Every n-variable polynomial $\mathbb{F}_q[x_1, \ldots, x_n]$ defines a function $\mathbb{F}_q^n \to \mathbb{F}_q$, and by counting truth tables, we see that there are $q^{(q^n)}$ of these.

Similarly, by counting monomials of degree less than q, there are $q^{(q^n)}$ elements in $\mathbb{F}_q[x_1, \ldots, x_n]/\langle x_1^q - x_1, \ldots, x_n^q - x_n \rangle$.

We defined a *Boolean model* as an n-tuple $(f_1, \ldots, f_n)$ of Boolean functions, where $f_j \colon \mathbb{F}_2^n \to \mathbb{F}_n$, and there are $(2^{2^n})^n = 2^{n2^n}$ of these. Analogously, an *algebraic model* (or "local model") is an n-tuple $(f_1, \ldots, f_n)$ of functions $\mathbb{F}_q^n \to \mathbb{F}_q$, and there are $(q^{q^n})^n = q^{nq^n}$. It is straightforward to see that there are $(q^n)^{q^n} = q^{nq^n}$ state space graphs, which means that every map $\mathbb{F}_q^n \to \mathbb{F}_q^n$ is the FDS map of an algebraic model. We covered the slides *Algebraic models and finite dynamics systems*, pp. 21-32 (mostly board work).

Suggested reading: Robeva/Macauley (2018), Chapter 4: *The Regulation of Gene Expression by Operons and the Local Modeling Framework*, Section 4.3.4, pp. 112–115.

- **WEEK 9: 10/14–10/18**.
  - **Mon. October 14**: Fall Break (no class)
  - **Wed. October 16**: We formalized the asynchronous automaton of an algebraic model $(f_1, \ldots, f_n)$, and showed that there are q^{nq^n} of them, which means that not only does every algebraic model define an asynchronous automaton, but each possible automaton arises from such a model. We formalized the notion of transient and periodic points for the synchronous state space. This is a little more complicated in the asynchronous case because of the possibility of complex attractors. In general, an attractor is a strongly connected component that has no outgoing edges from it. We covered the slides *Algebraic models and finite dynamics systems*, pp. 33-38.

    Suggested reading: Robeva/Macauley (2018), Chapter 4: *The Regulation of Gene Expression by Operons and the Local Modeling Framework*, Sections 4.3.5–4.3.6, pp. 115–118.
  - **Fri. October 18**: We formalized the notion of the wiring diagram of an algebraic model. Going forward, we will be particularly interested in *unate* functions, which are those whoe wiring diagrams have only positive or negative edges. We finished this set of slides by counting algebraic models and BNs with certain properties—this number grows *fast*!

    We discussed the well-known *network inference problems* in systems biology, and several general approaches, such as correlation networks, regression-based methods, information theoretical scores, Bayesian networks, ODE methods,

and algebraic methods, which we will be focusing on. We covered the slides *Algebraic models and finite dynamics systems*, pp. 39–45 and *The network inference problem*, pp. 1–3.

Suggested reading: One of the survey papers in the networkr inference problem, such as:

* M.M. Saint-Antoine & A. Singh (2020). Network inference in systems biology: recent developments, challenges, and applications. *Curr Opin Biotechnol.* **63**, 89–98.
* V.A. Huynh-Thu & G. Sanguinetti (2019). Gene regulatory network inference: an introductory survey. *Gene regulatory networks: Methods and protocols*, 1–23.

- **WEEK 10: 10/21–10/25**.

  – **Mon. October 21**: We began with an example of the state space of a Boolean model (f_1, f_2, f_3), with several edges missing, and asked "*which variables can depend on which variables"?* This is an inherently algebaric question. We argued why we can consider the coordinate functions individually, which us to posing this question for a function $f\colon \mathbb{F}_2^n \to \mathbb{F}_2$, given partial information. We considered a particular example, of a truth table that had 5 (of 8) missing values, and so 256 models that fit this data. Of them, only 4 are *unate*, which are the most "biologically meaningful." For each, we wrote the (signed) support, and illustrated which ones were minimal sets, called *min-sets*. Next, we outlined the broad goals of this section: (1) encode partial data algebraically with an ideal, (2) take its *primary decomposition* [analogue of factoring a number into prime powers], and (3) translate the primary components into minimal wiring diagrams.

  We finished with a brief discussion of *monomial ideals*, which are those generated by monomials. These algebraic objects have a strong combinatorial components, and we saw an example of this by drawing how a a monomial ideal is encoded by a *staircase diagram*. We covered the slides *The network inference problem*, pp. 4–11.

  Suggested reading: Robeva/Macauley (2018), Chapter 6: *Inferring Interactions in Molecular Networks via Primary Decompositions of Monomial Ideals*, Sections 6.1–6.2.1, pp. 175–182.

  – **Wed. October 23**: We explored the bijection between squarefree monomial ideals and simplicial complexes, which is Alexander duality. Specifically, if we associate (monic) squarefree monomials with subsets of $X = \{x_1, \ldots, x_n\}$, then monomials in I are closed under unions, and monomials not in I are closed

under intersections.

The *primary decomposition* of an ideal is a way to write $I = P_1 \cap \cdots \cap P_k$, where the P_i's is a *primary ideal*, called *primary components*. Loosely speaking, is the the "ideal analogue" of factoring an integer into prime powers: $200 = 2^3 \cdot 5^2$ in terms of ideals is $200\mathbb{Z} = 8\mathbb{Z} \cap 25\mathbb{Z}$. For squarefree monomial ideals, also called *Stanley-Reisner ideals*, the generators of the *primary components* of I are the complements of the maximal faces. We covered the slides *The network inference problem*, pp. 12–20.

Suggested reading: Robeva/Macauley (2018), Chapter 6: *Inferring Interactions in Molecular Networks via Primary Decompositions of Monomial Ideals*, Sections 6.2.2–6.2.3, pp. 182–191.

– **Fri. October 25**: We defined prime and primary ideals, and mentioned the Lasker-Noether theorem, which says that every ideal of a polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ (more generally, a Noetherina ring) has a primary decomposition. Though this need not be unique or easy to compute, square-free monomial ideals have a simple combinatorial description. After several examples, we applied the Stanley-Reisner theory (i.e., the theory of square-free monomial ideals) that we have built up to reconstructing algebraic models.

Given a set $\mathcal{D} = \{(\mathbf{s}_1, t_1), \ldots (\mathbf{s}_m, t_m)\}$ of "data" we say that a function f *fits the data* if $f(\mathbf{s}_i) = t_i$ for each i. The *model space* is the set of functions that fit the data. Given a set of data, we defined what it means for a set $\alpha \subseteq [n]$ to be *disposable* vs. *non-disposable*, and *feasible* vs. *infeasible*, with respect to a set of data. The non-disposable sets generated an ideal, and its primary components are the min-sets. We covered the slides *The network inference problem*, pp. 21–37.

Suggested reading: Robeva/Macauley (2018), Chapter 6: *Inferring Interactions in Molecular Networks via Primary Decompositions of Monomial Ideals*, Sections 6.3, pp. 191–199.

• **WEEK 11: 10/28–11/1**.

– **Mon. October 28**: We saw how to modify the previous construction of the ideals of non-disposable sets to the ideal of signed non-disposable sets. Specifically, instead of encoding the coordinates in which $\mathbf{s}_i$ and $\mathbf{s}'_i$ differ with a monomial $m(\mathbf{s}_i, \mathbf{s}_j)$, we use a pseudomonomial $p(\mathbf{s}_i, \mathbf{s}_j)$, and use $\overline{x_i}$ if the interaction is negative. As before, the primary components give the signed min-sets, but only for functions that are unate. We saw an a published example, involving

the gene network for mesodermal tissue development in the model organism (worm) *C. elegans*.
Finally, we saw how recent published work (Harrington et al.) has extended these ideas to continuous-space dynamical systems $f\colon [0,1]^n \to [0,1]^n$. Several examples included difference equation models of flour beetles and fish. We covered the slides *The network inference problem*, pp. 38–48.

Suggested reading:

- Robeva/Macauley (2018), Chapter 6: *Inferring Interactions in Molecular Networks via Primary Decompositions of Monomial Ideals*, Sections 6.4–6.6, pp. 199–210.
- Harrington, H. A., Stillman, M., & Veliz-Cuba, A. (2024). Algebraic network reconstruction of discrete dynamical systems. *Adv. Appl. Math.*, **161**, 102760.
- Veliz-Cuba, A., Newsome-Slade, V., & Dimitrova, E. S. (2024). A unified approach to reverse engineering and data selection for unique network identification. *SIAM J. Appl. Dyn. Syst.* **23**(1), 592-615.

– **Wed. October 30**: We begin with the same "partial state space" of a 3-node Boolean model as in the last section, but this time asked: "*what can we say about the set of all functions that fits this data.*" A simple answer is simply just all ways to fill out the truth table, but that doesn't elucidate the algebraic structure. As before, we can consider each function $f_j\colon \mathbb{F}_2^n \to \mathbb{F}_2$ separately.

Using the same example as the one from last week (which had min-sets $\{x_1, \overline{x_3}\}$ and $\{x_1, \overline{x_2}\}$), we first attempted to construct a single funtion. This was done by polynomial interpolation, in a similar manner to how this is taught with Chebyshev polynomail interpolation in our numerical analysis (Math 8600), dual bases of a vector space of polynomials in linear algebra (Math 8530), and the proof of the Chinese remainder theorem in abstract algebra (Math 8510). The general idea is to construct a polynomial $r_i(x)$ such that $r_i(\mathbf{s}_i) = 1$ and $r_i(\mathbf{s}_j) = 0$ for $i \neq j$, and then to use $f(x) = t_1 r_1(x) + \cdots + t_k r_k(x)$. We covered the slides *The model space*, pp. 1–8.

Suggested reading: Robeva/Hodge (2013), Chapter 3: *Inferring the Topology of Gene Regulatory Networks: An Algebraic Approach to Reverse Engineering*, by E. Dimitrova and B. Stigler. Sections 3.1–3.3, pp. 75–85.

– **Fri. November 1**: Given a set $\mathcal{D} = \{(\mathbf{s}_1, t_1), \ldots, (\mathbf{s}_k, t_k)\}$ of data, the *vanishing ideal* is the set $I(\mathcal{D})$ of (polynomial) functions f for which $f(\mathbf{s}_i) = 0$ for all i. The *model space* of $\mathcal{D}$ is the set of functions which fit the data, i.e., $f(\mathbf{s}_i) = t_i$. This is a coset of the vanishing ideal, i.e., $\mathrm{Mod}(\mathcal{D}) = f + I(\mathcal{D})$.

We have seen this structure elsewhere, as the equation of a line or plane not through the origin, or the solution set to $Ax = b$, or to an inhomogeneous linear ODE. We covered the slides *The model space*, pp. 8–13.

Suggested reading: Robeva/Hodge (2013), Chapter 3: *Inferring the Topology of Gene Regulatory Networks: An Algebraic Approach to Reverse Engineering*, by E. Dimitrova and B. Stigler. Section 3.4, pp. 85–88.

- **WEEK 12: 11/4–11/8**.
  - **Mon.–Wed November 4–6**: We saw how to compute the vainishing ideal $I(\mathcal{D})$ of a set $\mathcal{D} = \big\{(\mathbf{s}_1, t_1), \ldots, (\mathbf{s}_k, t_k)\big\}$ of data, as the intersection of the ideals that vanish on the individual input vectors. Though it is not easy to find a basis of such an intersection by hand, Macaulay2 can do it right away. Next, we saw how to find the model space of a set $\mathcal{D} = \big\{(\mathbf{s}_1, \mathbf{t}_1), \ldots, (\mathbf{s}_k, \mathbf{t}_k)\big\}$ of data of input vector and output vectors. This is done by applying the previous method to the individual coordinates. We did this with an example of a 3-node Boolean model, where 3 of the 8 edges are unknown. The resulting model space has $8^3 = 512$ functions, and is

$$\mathrm{Mod}(\mathcal{D}) = \big[f_1 + I(\mathcal{D})\big] \times \big[f_2 + I(\mathcal{D})\big] \times \big[f_3 + I(\mathcal{D})\big],$$

    where the vanishing ideal $I(\mathcal{D})$ has size 8. We covered the slides *The model space*, pp. 14–23.

    Suggested reading: Robeva/Hodge (2013), Chapter 3: *Inferring the Topology of Gene Regulatory Networks: An Algebraic Approach to Reverse Engineering*, by E. Dimitrova and B. Stigler. Sections 3.5–3.6, pp. 89–100.

  - **Fri. November 8**: We finished Part III with how to reverse-engineer the models space of a (non-Boolean) algebraic network, using an example of a 3-node model over $\mathbb{F}_5$. Then, we discussed 1D dynamical system maps, and the logistic map $x_{n+1} = rx_n(1 - x_n)$. Of particular importance was to see how the dynamics changes as a function of the parameter r. This system has a stable (overdamped) fixed point for $0 \leq r \leq 2$, a stable (underdamped) fixed point for $2 \leq r \leq 3$. For $3 < r < 3.45$, the is a size-2 stable attractor, followed by a size-4, then size-8 attractor, and so on, until the onset of chaos, at $r \approx 3.56995$. Even for some larger values, there are islands of stability, like a 3-cycle for $r \approx 3.83$. Though these can be studied analytically, we mostly used Matlab to illustrate the main ideas, especially using cobwebbing diagrams. We covered the slides *The model space*, pp. 24-28, and *Random Boolean networks*, pp. 1–6.

Suggested reading: R.M. May, (1976). Simple mathematical models with very complicated dynamics. *Nature* **261**(5560), 459–467.

- **WEEK 13: 11/11–11/15**.

  - **Mon. November 11**: We started the day with a quick video and animation to see how the Mandlebrot set appears in the bifurication diagram of the logistic map. Then, we gave a brief summary of what statistical mecanics is: the study of large assemblies of microscope entities using the tools of probability and statistics. In 1969, Stuart Kauffman introduced random Boolean networks (RBNs) as models of gene regulatory networks. In his model, there are N nodes, each having K randomly chosen inputs. In other words, the network topology is construct first. Next, the functions $f_i \colon \mathbb{F}_2^K \to \mathbb{F}_2$ are assigned, via some distribution. These RBNs are called *NK-networks*. Kauffman noticed that for $K = 1$, the RBNs had small attractors and lots of fixed points, and perturbations tended to die out. These networks are *stable*. For large K, the RBNs had lots of large attractors, and perturbations tended to propogate. These networks are called *chaotic*. Networks for $K = 2$ are on the boundary of these phases, and are called *critical*. Early evidence seemed to suggest that biological networks shared a number of properties (e.g., scaling laws) with these random critical networks.

    In 1942, geneticist C.H. Waddington developed the concept of an epigenetic landscape, and *canalization*, a measure of evolution robustness that quantifies how a population can produced the same phenotype despite changes to its environment or genotype. This was quantified in Boolean functions by Stuart Kauffman in 1993. A Boolean function $f \colon \mathbb{F}_2^n \to \mathbb{F}_2$ is *canalizing* if for some variable i, taking an input value $x_i = a$ completely determines the output. These are precisely functions of the form $f(x_1, \ldots, x_n) = y_i \Diamond g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, where $y_i \in \{x_i, \overline{x_i}\}$ and $\Diamond \in \{\wedge, \vee\}$. Most functions in Boolean models are canalizing, and it's been shown the RBNs built with them are more stable.

    One way to assigned the functions in a RBN is to pick Boolean functions with *bias p*, which measn that the truth table is a length-2^K vector of iid Bernoulli random variables. A uniform distribution is the special case of $p = 1/2$. Other distributions include weighted classes, only picking canalizing functions, or picking all of the same function (e.g., only "AND functions"). We covered the slides *Random Boolean networks*, pp. 7–14.

    Suggested reading: B. Drossel, (2008). Random Boolean networks. *Reviews of nonlinear dynamics and complexity*, Sections 1 & 2, (pp. 1–5 of `https:`

//arxiv.org/pdf/0706.3351).

– **Wed. November 13**: We started by talking about percolation theory, and compared square lattices where each edge exists with probability p. If $p = 1/4$, then this picture looks like "islands." If $p = 3/4$, then it looks like "holes," and if $p = 1/2$, then it's somewhere between these two phases. This "phase transition" is an important topic in statistical mechanics, and we'll see something similar with RBNs.

In the context of random Boolean networks, we defined what it meant for a probability distribution on functions to be *permutation-invariant* (e.g., biased functions), and *inversion-invariant* (e.g., weighted classes). A straightforward probabilistic argumente shows that if an inversion-invariant distribution is used, then the expected number of fixed points will be 1. A little longer argument established the same result for biased functions. Finally, we summarized an argument that gave the expected number of length-L cycles in a RBN with $K = 2$, which was used to show that the number of attractors in a $K = 2$ network grows faster than any power law. We covered the slides *Random Boolean networks*, pp. 14–19.

Suggested reading: B. Drossel, (2008). Random Boolean networks. *Reviews of nonlinear dynamics and complexity*, Section 3, (pp. 5–9 of `https://arxiv.org/pdf/0706.3351`).

– **Fri. November 15**: We discussed how to measure the growth of a small perturbation near a steady-state P^* in a difference equation: plug in $P_t = P^* + p_t$ and $P_{t+1} = P^* + p_{t+1}$, disregard non-linear terms, and solve for $|p_{t+1}/p_t|$. If this is less than 1, then P^* is stable; otherwise it is unstable. Alternatively, if $P_{t+1} = F(P_t)$, then the dynamics are stable if $|F'(x)| < 1$ and chaotic if $|F'(x)| > 1$.

We discussed the Derrida and Pomeau's *annealed approximation* of random Boolean networks. The network is assumed to be infinitely large, so flucations of global quantities are negligible, and the inputs of each node are reset at each time-step. In many case, the time-evolution of weight (i.e., the proportion of 1s) can be derived as a 1D dynamical system, $b_{t+1} = F(b_t)$, and we saw this using biased, threshold, and canalizing functions. We saw examples of where this statistic is both ordered and chaotic. We covered the slides *Random Boolean networks*, pp. 19–25.

Suggested reading: B. Drossel, (2008). Random Boolean networks. *Reviews of nonlinear dynamics and complexity*, Sections 4–7, (pp. 9–19 of `https:`

//arxiv.org/pdf/0706.3351).

- **WEEK 14: 11/18–11/22**.

  – **Mon. November 18**: We defined the *Boolean* j^{th} *partial derivative*, which is 1 if toggling the j^{th} bit flips the output, and 0 otherwise. The expected value of this across all 2^K inputs is the *activity* of x_j in f, denoted α_j^f. This is simply the probability that toggling the j^{th} bit flips the output. The *sensitivity* $s^f(x)$ of f at x is the number of Hamming neighbors on which f is different. The *average sensitivity* is just the sum of the activities: $s^f = \alpha_1^f + \cdots + \alpha_K^f$.

  In a general dynamical system, the *Lyapunov exponent* measures the divergence rate of infinitesimally close trajectories. Small perturbations grow if $\lambda > 0$ and they shrink of $\lambda < 0$. The Boolean network analogue of this is the logarithm of the average sensitivity. For a random Boolean function f with bias p, the average sensitivity is $2Kp(1-p)$. This defines the *critical threshold* between stable networks and chaotic networks, for $p = 1/2$. We covered the slides *Random Boolean networks*, pp. 26–33.

  Suggested reading: Shmulevich, Ilya, and Stuart A. Kauffman. Activities and sensitivities in Boolean network models. *Phys. Rev. Lett.* **93**.4 (2004): 048701.

  – **Wed. November 20**: We said a few works about the concept of entropy from information theory, and watched a few minutes of 3blue1brown's wonderful video on Wordle and information theory. The concept of *basin entropy* of Boolean networks has been studied, and was shown to be maximized for critical networks.

  We moved onto a different topic: biological feedback. In 1981, René Thomas observed that positive feedback loops (in the wiring diagram) seemed to be necessary for multistationarity, and negative feedback seemed to be necessary for sustained oscillations. This has since been explored and (mostly) proven in a variety of frameworks, including differential equations to Boolean and logical models. We stated two versions of Rule 1 in an ODE framework, from 2002 and 2003, respectively. We stated two partial results for Rule 2, from 1998.

  Moving onto the Boolean model framework, we defined the *discrete* j^{th} *partial derivative* at $x \in \mathbb{F}_2^n$, denoted $f_{ij}(x) = \partial f_i(x)/\partial x_j \in \{-1, 0, 1\}$, which tells us whether increasing the j^{th} bit $0 \to 1$ increases, decreases, or preserves the output. The *local wiring diagram* $Gf(x)$ is the graph on $\{1, \ldots, n\}$ with a signed edge from j to i for each nonzero $f_{ij}(x)$. The (global) wiring diagram is simply the union of these local wiring digrams. We did this for an example Boolean

model, and saw how this resolves the ambiguity of the signs of interactions when a varable x_i and its negation $\overline{x_i}$ both appear in the logical expression. We covered the slides *Biological feedback*, pp. 1–6.

Suggested reading: Read the section *References to more recent work* in the Preface (pages 7–13) of the 2006 update to the book: Thomas, R. and D'Ari, T. Biological Feedback. *CRC Press*, 1990. This can be found online. Also, watch the entire 3blue1brown video on Wordle and information theory.

– **Fri. November 22**: After taking HW questions, we reviewed the concept of a local wiring diagram, constructed from discrete partial derivatives. Contrapositives of Thomas' rules say that if the wiring digram has no positive (reps., negative) cycle, then f has at most (resp., at least) one fixed point. As a corollary, if the wiring diagram is acyclic, then it has a unique fixed point. In fact, we can say more (Robert, 1980): f^n is constant, and the asynchronous automaton $\mathcal{A}(f)$ is acyclic and has a geodesic path from every state to the unique fixed point. A Boolean model for which f^k is contstant for some k is said to be *nilpotent*. In 2019, A. Richard studied nilpotent dynamics, and proved some partial converses to Thomas' rules.

The *Jacobian conjecture* in algebraic geometry is a famous open problem from 1939. It says that (assuming $\operatorname{char} F = 0$) if $f\colon \mathbb{F}^n \to \mathbb{F}^n$ is a polynomial map $f\colon x \mapsto (f_1(x), \ldots, f_n(x))$, then the $\det J$ is a non-zero constant function iff f is invertible (the "$\Leftarrow$" direction is trivial). In 1999, an equivalent statement was proven, saying that if every eigenvalue of the Jacobian lies withing the unit circle, for all $x \in \mathbb{F}^n$, then f has a unique fixed point. The "Boolean analogue" of this was proven by Shih & Dong in 2005: if every eigenvalue of a Boolean network is zero, then f has a unique fixed point, which is a stronger version of Robert's theorem. We covered the slides *Biological feedback*, pp. 6–11.

Suggested reading: Richard, A. (2019). Positive and negative cycles in Boolean networks. *J. Theor. Biol.*, **463**, 67-76.

- **WEEK 15: 11/25–11/29**.

– **Mon. November 25**: We summarized the main results from the last 25 years in the literature about Thomas' rules on biological feedback. We also saw two explicit examples: Boolean models when the wiring diagram is a simple chordless positive cycle, and a simple chordless negative cycle. The first example has two fixed points, and the second example has a length-$2n$ cycle. We will finish the class next week by proving the Boolean model version of Thomas' second rule, and we ended this class with some definitions, and an outline of

how the proof will go. We covered the slides *Biological feedback*, pp. 11–21.

Suggested reading: Richard, A. (2010). Negative circuits and sustained oscillations in asynchronous automata networks. Adv. Appl. Math. 44(4), 378-392.

– **Wed.–Fri. November 27–29**: Thanksgiving Break (no class)

- **WEEK 16: 12/2–12/6**.

– **Mon. December 2**: We finished the proof of Robert's second rule, by A Richard (2010). One key idea is the *strong local wiring diagram* $G'f(x)$, which is a subgraph of the ordinary local wiring diadgram $Gf(x)$ (Lemma 1). Next, Lemma 2 describes a condition of when the local wiring diagram will have a path from j to i: if the path starts with "edge j", and ends with the first time that "edge i flips." Lemma 3 says that if the asynchronous automaton has cyclic attractor A and at least one node has out-degree 1, then the strong wiring diagram has a negative cycle. This is done by applying Lemma 2 in the special case of $i = j$, because in every negative cycle, there is path starting and ending with an "edge i," of opposite signs.

In Lemma 4, this is proven without the "out-degree 1" condition, by showing that for some i, replacing $f_i(x)$ with the identity function results in a strictly smaller attractor $A' \subsetneq A$. This puts a partial order on the pair (f, A) of Boolean models with attractors. Induction over this poset shows that the strong wiring diagram has a negative cycle, which means that the wiring diagram must as well. We covered the slides *Biological feedback*, pp. 21–38.

Suggested reading: Feliu, E., & Wiuf, C. (2015). Finding the positive feedback loops underlying multi-stationarity. *BMC Syst. Biol.* 9, 1-12.

– **Wed. December 4**: In-class exam.

– **Fri. December 6**: Class presentations (to be continued during final's week).