Biological Computation

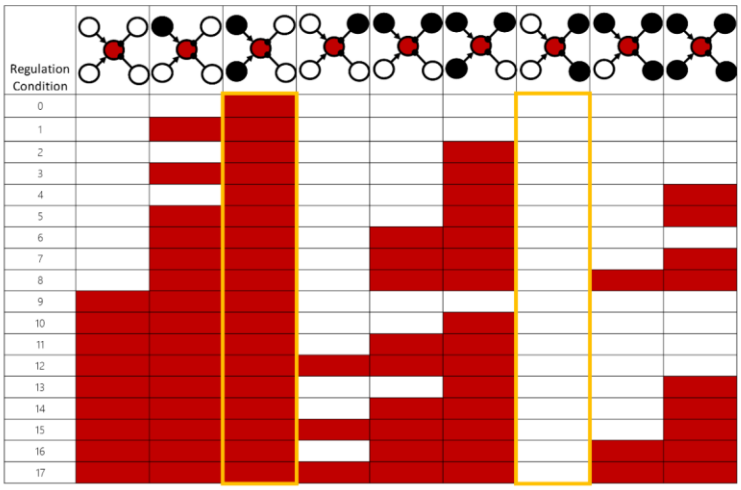
Final Project

Ziv Chaba 326681285

Bar Zomer 326400454

https://github.com/Ziv33/BiologicalComputation

First, based on the definition, we wrote a computer program in python that finds all the monotonic regulation conditions of the reasoning engine as we saw in class, and appear in the following table:



Regulation Condition									
0			🟥						
1		🟥	🟥						
2			🟥			🟥			
3		🟥	🟥			🟥			
4			🟥			🟥			🟥
5		🟥	🟥			🟥			🟥
6		🟥	🟥		🟥	🟥			
7		🟥	🟥		🟥	🟥			🟥
8		🟥	🟥		🟥	🟥		🟥	🟥
9	🟥	🟥	🟥						
10	🟥	🟥	🟥			🟥			
11	🟥	🟥	🟥		🟥	🟥			
12	🟥	🟥	🟥	🟥	🟥	🟥			
13	🟥	🟥	🟥			🟥			🟥
14	🟥	🟥	🟥		🟥	🟥			🟥
15	🟥	🟥	🟥	🟥	🟥	🟥			🟥
16	🟥	🟥	🟥		🟥	🟥		🟥	🟥
17	🟥	🟥	🟥	🟥	🟥	🟥		🟥	🟥

All the monotonic regulation conditions of the reasoning engine

In the table, there are nine columns which corresponds to all the nine configurations of basic systems with maximum of two activators and two inhibitors. As discussed in class, there can be 2^9 forms of regulation conditions with this setup of nine basic systems, but the monotonic requirement reducing this number drastically.

We stated by finding the basic regulation condition (or, functions) for each one of these nine basic system, those which are given immediately by the definition- Given IN, the maximum number of inhibitors in the system, and AC, the maximum number of activators in the system, then, if for a pair $(\#\, inhibitors, \#\, activators) = (n_1, m_1)$, its target is 1, then for every $[n_2 = n_1]$ & $[m\_1 < m\_2 <= AC]$ and every $[n_2 < n_1]$ & $[m_2 >= m_1]$, the pairs (n_2, m_2) must have $target = 1$.

We also acknowledged the assumption which we saw in class, and also appears in the "*Supplementary Material*" attachment in the ninth section in the course site, that for each regulation condition, the component cannot be constantly activated or repressed regardless of the state of each regulators, and therefore states $(n = IN, m = 0, target = 1), (n = 0, m = AC, target = 0)$ (respectively) are forbidden. This assumption reflects in the highlighted columns in the above table.

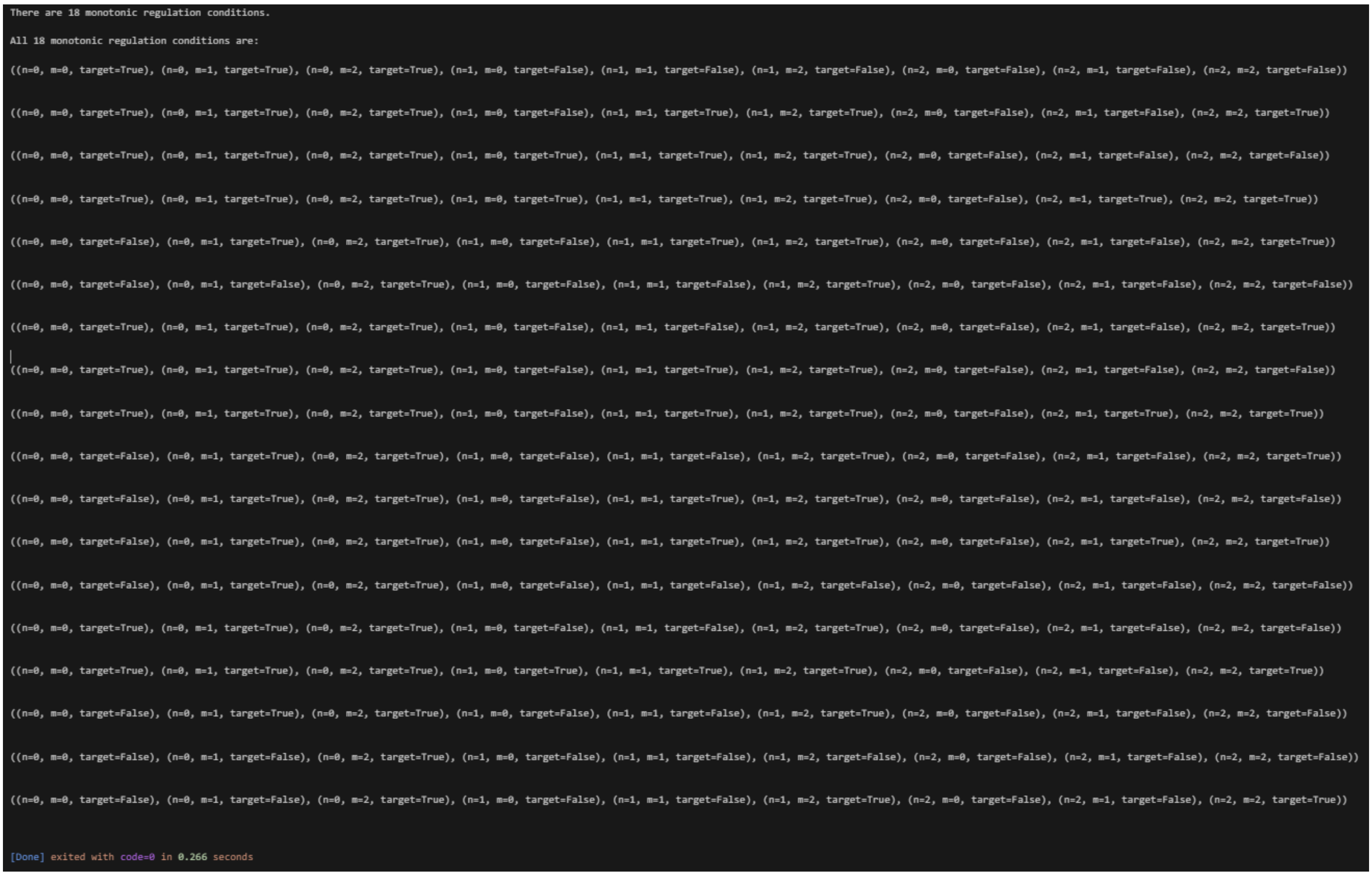
Since we are only considering scenarios where none, some, or all activators or inhibitors are available to a target, it is sufficient to illustrate cases with up to two regulators of each type, as shown in the table above. In our code, we generalized this approach using the variables AC and IN, which represent the number of activators and inhibitors, respectively. By adjusting these variables, the code can generate and display the corresponding set of monotonic regulation conditions for any given number of activators and inhibitors.

Subsequently, with those basic monotonic regulation conditions, we expanded and complete our set of monotonic regulation conditions by applying monotonicity-preserving operations for Boolean functions - AND, OR between the basic monotonic regulation conditions (functions) that we have already found.

We started with a set of the basic monotonic regulation conditions, and in every loop we commit AND and OR operations between each of the target bit between the same basic system (among the nine) in the two basic monotonic regulation conditions (in terms of data structures view, it is AND, OR operations bitwise among a tuple with length of nine), and we add this new monotonic condition to the final set. We stop the process when the set in no more changed. As mentioned, the operations AND, OR are monotonicity-preserving operations for Boolean functions, as we saw in previous courses, by applying them on monotonic regulation functions (conditions) we are still getting monotonic functions (condition), so in this way we guarantee to get no more conditions than the actual monotonic regulation condition set. Because we started to perform the AND, OR operations from a starting point of the minimal set of monotonic regulation conditions, this set is actually spanning all the options for monotonic conditions, meaning it is also guarantee for us to get not lower conditions than we supposed to get, and altogether we manage to get the all and only monotonic regulation conditions of the reasoning engine.

The program contains the BasicSystem.py file, which holds the struct of each of the basic systems, and main,py file which integrates all the methods into a comprehensive algorithm.

The output of the program is as follows:



```
There are 18 monotonic regulation conditions.

All 18 monotonic regulation conditions are:

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=False), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=True))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=True), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=True), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=True), (n=2, m=2, target=True))

((n=0, m=0, target=False), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=True))

((n=0, m=0, target=False), (n=0, m=1, target=False), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=True))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=True), (n=2, m=2, target=True))

((n=0, m=0, target=False), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=True))

((n=0, m=0, target=False), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=False), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=True), (n=2, m=2, target=True))

((n=0, m=0, target=False), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=False), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=True), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=True), (n=1, m=1, target=True), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=True))

((n=0, m=0, target=False), (n=0, m=1, target=True), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=False), (n=0, m=1, target=False), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=False), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=False))

((n=0, m=0, target=False), (n=0, m=1, target=False), (n=0, m=2, target=True), (n=1, m=0, target=False), (n=1, m=1, target=False), (n=1, m=2, target=True), (n=2, m=0, target=False), (n=2, m=1, target=False), (n=2, m=2, target=True))


[Done] exited with code=0 in 0.266 seconds
```

We consolidated these results into the following table:

Monotonic Regulation Condition Number In The Given Table	Monotonic Regulation Condition Number In Our Code's Printout	(n=0 m=0)	(n=0 m=1)	(n=0 m=2)	(n=1 m=0)	(n=1 m=1)	(n=1 m=2)	(n=2 m=0)	(n=2 m=1)	(n=2 m=2)
0	16			TRUE						
1	12		TRUE	TRUE						
2	5			TRUE			TRUE			
3	15		TRUE	TRUE			TRUE			
4	17			TRUE			TRUE			TRUE
5	9		TRUE	TRUE			TRUE			TRUE
6	10		TRUE	TRUE		TRUE	TRUE			
7	4		TRUE	TRUE		TRUE	TRUE			TRUE
8	11		TRUE	TRUE		TRUE	TRUE		TRUE	TRUE
9	0	TRUE	TRUE	TRUE						
10	13	TRUE	TRUE	TRUE			TRUE			
11	7	TRUE	TRUE	TRUE		TRUE	TRUE			
12	2	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE			
13	6	TRUE	TRUE	TRUE			TRUE			
14	1	TRUE	TRUE	TRUE		TRUE	TRUE			TRUE
15	14	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE			TRUE
16	8	TRUE	TRUE	TRUE		TRUE	TRUE		TRUE	TRUE
17	3	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE		TRUE	TRUE

Overall, by applying this algorithm, we showed that the 18 rows in the given table are indeed correspond to the only regulation conditions that satisfy monotonic requirement and consider whether none, some or all of the activators / inhibitors are present.

Repository of logically consistent real-world Boolean network models

Overview

Boolean network models play a crucial role in systems biology by modeling complex biological interactions using binary states and logical rules. Historically, creating these models required extensive expert input and manual adjustments. With advances in computational methods, there is now a strong need for a comprehensive, reliable repository of Boolean network models to aid in the validation and benchmarking of new computational tools.

The study "Repository of logically consistent real-world Boolean network models " presents the Biodivine Boolean Models (BBM) dataset, an open-access collection of over 230 Boolean network models. These models, sourced from various databases and scientific literature, provide a standardized benchmark essential for evaluating new computational methodologies in systems biology.

Key Benefits of the BBM Dataset

The BBM dataset offers several notable advantages:

- **Multiple Formats**: The models are available in various commonly used formats, including bnet, aeon, and sbml. This variety ensures compatibility with different analytical tools, avoiding the need for format conversions.
- **Standardized Representation**: The dataset standardizes model representation and resolves ambiguities, especially concerning the treatment of input nodes. This uniformity ensures consistent analysis and interpretation across studies.
- **Thorough Validation**: A significant feature of the BBM dataset is its rigorous validation process. Each model undergoes a thorough check for logical consistency, addressing issues such as discrepancies between regulation and update functions. This process ensures that the models are both accurate and reliable.
- **Customizable Options**: The dataset allows researchers to generate custom subsets based on specific criteria, making it easier to select models that align with particular research needs.

Creation and Maintenance of the BBM Dataset

The BBM dataset was developed through a structured approach:

- **Collection**: Models were gathered from reputable databases and academic sources. The dataset includes any Boolean model representing a real biological system, regardless of prior curation.
- **Normalization and Validation**: Each model was assigned a unique identifier and normalized for consistency. This process involved standardizing variable names and verifying input nodes. Models were validated for logical coherence, with any inconsistencies corrected and documented.
- **Ongoing Updates**: Managed through a version-controlled git repository, the BBM dataset benefits from continuous updates and maintenance. Official releases provide reliable references for researchers.

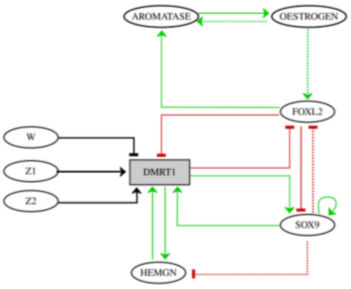
Conclusion

The BBM dataset represents a significant advancement in systems biology by offering a large, standardized collection of logically consistent Boolean network models. As a major resource for researchers, it enhances the reproducibility and accuracy of computational analyses, supporting the development of effective methods for studying complex biological systems.

We checked out the github repository and the Boolean networks which provided and we selected the model Chicken-Sex-Determination-Reduced which is described in the paper "Logical Modelling Uncovers Developmental Constraints for Primary Sex Determination of Chicken Gonads" by Lucas Sánchez and Claudine Chaouiya.

Overview

This article by Sánchez and Chaouiya provides a significant contribution to the understanding of sex determination in chickens. Utilizing a logical modeling approach, the authors assemble current biological knowledge and offer insights into the regulation of primary sex determination in chickens. They construct a gene regulatory network based on experimental data and prevailing hypotheses, leading to a comprehensive logical model that integrates both the Z-dosage and dominant W-chromosome hypotheses:



The primary focus of the model is the mutual inhibition between DMRT1, a gene linked to the Z chromosome, and FOXL2, a gene involved in ovarian development. The study finds that the initial amount of DMRT1 product plays a pivotal role in determining the sexual fate of the gonads. In this regard, a W-linked factor is suggested to act as a secondary mechanism by reducing DMRT1 levels in ZW gonads, thereby influencing the resolution of sexual fate. This W-factor serves to reinforce the difference in DMRT1 levels between ZW and ZZ gonads, emphasizing its role at the initiation step of sex determination.

Furthermore, the model uncovers several developmental constraints crucial for sex determination. These constraints define qualitative restrictions on the relative transcription rates of genes like DMRT1, FOXL2, and HEMGN. For instance, the model suggests that the activation of FOXL2 should be slower than that of HEMGN and DMRT1, ensuring a successful progression towards the intended sexual phenotype. This insight into the timing and sequence of gene expression is a critical aspect of the study, as it highlights the importance of temporal regulation in developmental processes.

The model also explores the role of additional genes such as SOX9, which is crucial for the maintenance of DMRT1 expression and thereby the development of testes. SOX9's auto-regulatory loop, which maintains DMRT1 at high levels, underscores its role in sustaining male sexual development. The inclusion of SOX9 demonstrates how the interaction of various genetic components can lead to the stabilization of a chosen sexual pathway.

Another significant contribution of the study is the use of logical modeling to simulate various mutant conditions. By applying loss-of-function (LOF) and gain-of-function (GOF) mutations, the authors demonstrate how alterations in key genes can lead to changes in sexual phenotype, such as the transformation of ZZ gonads to an ovary phenotype or ZW gonads to a testis phenotype. These simulations provide a robust framework for predicting the outcomes of genetic manipulations and offer a valuable tool for further experimental studies.

Shortcoming

While the article presents a comprehensive model for primary sex determination in chickens, it overlooks the variability in natural populations. The model relies on predefined genetic interactions and assumes consistency across all chicken populations, neglecting the influence of natural genetic diversity and environmental factors, such as temperature and epigenetic modifications, that can affect gene expression. Although the study successfully models key genes like DMRT1, FOXL2, and HEMGN, it may not capture all genetic factors influencing sex determination across avian species. Future research could enhance the model's robustness by incorporating a wider range of genetic and environmental factors, improving its predictive power.

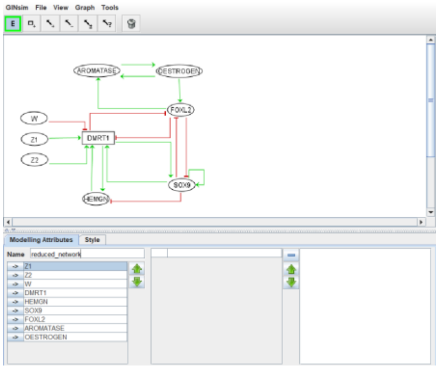
Impressions

Overall, our impression of the article is positive. The study provides a thorough and innovative approach to understanding the complex process of sex determination in chickens. By utilizing logical modeling, the authors offer a valuable framework that integrates existing knowledge and generates testable predictions. This approach not only clarifies the roles of specific genes like DMRT1 and FOXL2 but also provides insights into the dynamic interactions that drive sexual differentiation. Despite the above limitations, the study represents a significant step forward in the field of developmental biology and offers a strong foundation for future research.

The use of simulations to explore the outcomes of various genetic mutations is particularly commendable. These simulations help illustrate how specific genetic changes can influence sexual development, offering practical implications for experimental research. Additionally, the model's ability to integrate both the Z-dosage and W-factor hypotheses provides a more comprehensive understanding of chicken sex determination than models that consider these hypotheses in isolation.

Reproducing The Main Results

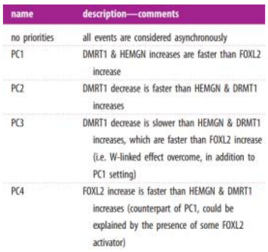
In this study, the authors used GINsim, a software tool for logical modeling, to simulate the gene regulatory network involved in the primary sex determination of chicken gonads. These simulations aimed to clarify the roles of key genes such as DMRT1, FOXL2, HEMGN, and SOX9 in determining whether the bipotential gonads in chickens develop into testes (male) or ovaries (female).



Name: reduced_network

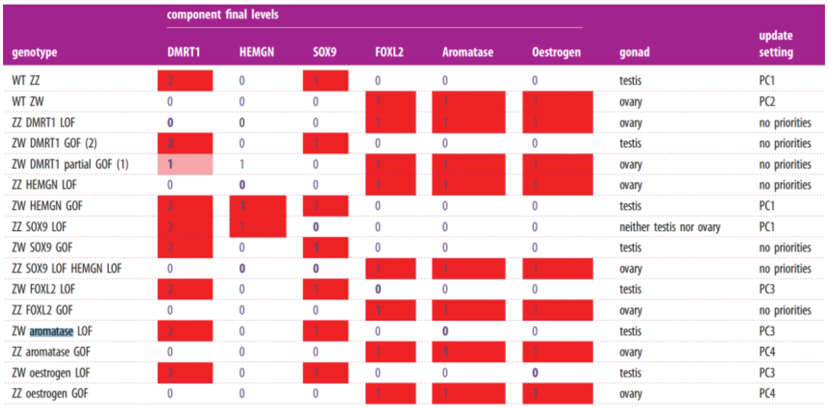
Our Model in the GINsim Tool

The gene regulatory network was assembled based on experimental data and existing hypotheses. The model incorporated both Z-linked and W-linked genes to simulate the processes that guide the development of male (ZZ) and female (ZW) gonads. Simulations were performed using asynchronous updating schemes, with specific priority settings to resolve conflicts in gene expression dynamics, as described in the article:



name	description—comments
no priorities	all events are considered asynchronously
PC1	DMRT1 & HEMGN increases are faster than FOXL2 increase
PC2	DMRT1 decrease is faster than HEMGN & DRMT1 increases
PC3	DMRT1 decrease is slower than HEMGN & DMRT1 increases, which are faster than FOXL2 increase (i.e. W-linked effect overcome, in addition to PC1 setting)
PC4	FOXL2 increase is faster than HEMGN & DMRT1 increases (counterpart of PC1, could be explained by the presence of some FOXL2 activator)

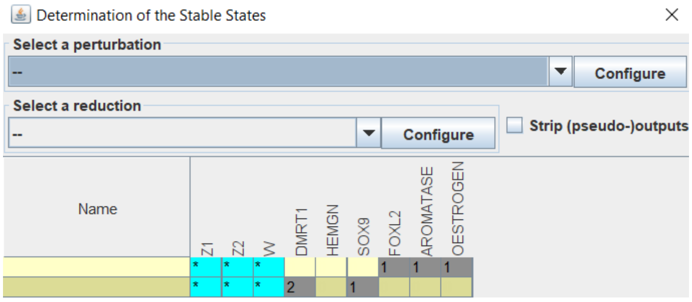
As mentioned, these update settings are already activated by the model. The main experiments involved simulating both wild-type and mutant gonads. Mutant conditions included loss-of-function (LOF) and gain-of-function (GOF) mutations of critical genes. These simulations were analyzed to determine the stable states of the network and the corresponding sexual phenotypes (testis or ovary). the authors considered the following criteria for defining the sexual phenotypes: expression of DMRT1 and SOX9 and absence of FOXL2, AROMATASE and OESTROGEN indicate testis identity, while expression of FOXL2, AROMATASE and OESTROGEN and absence of DMRT1 and SOX9 indicate ovary identity. The main results are concentrated in *Table 2*:



	component final levels							
genotype	DMRT1	HEMGN	SOX9	FOXL2	Aromatase	Oestrogen	gonad	update setting
WT ZZ	2	0	1	0	0	0	testis	PC1
WT ZW	0	0	0	1	1	1	ovary	PC2
ZZ DMRT1 LOF	**0**	0	0	1	1	1	ovary	no priorities
ZW DMRT1 GOF (2)	**2**	0	1	0	0	0	testis	no priorities
ZW DMRT1 partial GOF (1)	**1**	1	0	1	1	1	ovary	no priorities
ZZ HEMGN LOF	0	**0**	0	1	1	1	ovary	no priorities
ZW HEMGN GOF	2	**1**	1	0	0	0	testis	PC1
ZZ SOX9 LOF	2	1	**0**	0	0	0	neither testis nor ovary	PC1
ZW SOX9 GOF	2	0	**1**	0	0	0	testis	no priorities
ZZ SOX9 LOF HEMGN LOF	0	**0**	**0**	1	1	1	ovary	no priorities
ZW FOXL2 LOF	2	0	1	**0**	0	0	testis	PC3
ZZ FOXL2 GOF	0	0	0	**1**	1	1	ovary	no priorities
ZW aromatase LOF	2	0	1	0	**0**	0	testis	PC3
ZZ aromatase GOF	0	0	0	1	**1**	1	ovary	PC4
ZW oestrogen LOF	2	0	1	0	0	**0**	testis	PC3
ZZ oestrogen GOF	0	0	0	1	1	**1**	ovary	PC4

The data of this table was created by the GINsim model, and we reproduced all the main results:

W-Linked Gene Reduces DMRT1 Levels in ZW Gonads:
The W-linked gene acts to reduce DMRT1 levels in ZW gonads, favoring ovary development. Table 2 shows that **wild-type (WT) ZW gonads** develop into ovaries due to lower DMRT1 levels influenced by the W-linked gene. Below there is a figure of the default phase of the computation of stable states in the GINsim model, which correspond to the first two lines of *Table 2*, when the first relates to the ZW and the second to the ZX.



Select a perturbation: --

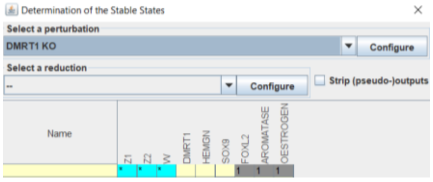
Select a reduction: --

☐ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*				1	1	1
	*	*	*	2		1			

Mutual Inhibition Between DMRT1 and FOXL2 Determines Sexual Fate:
The model shows that the sexual fate of chicken gonads depends on the mutual inhibition between DMRT1 and FOXL2. Higher initial levels of

DMRT1 in ZZ gonads promote testis development, while lower levels in ZW gonads lead to ovary development. Table 2 reflects this in the **loss-of-function (LOF) mutation in DMRT1** in ZZ gonads, which results in an ovary phenotype, demonstrating the necessity of DMRT1 for male development. Conversely, a **gain-of-function (GOF) mutation in DMRT1** in ZW gonads results in a testis phenotype, confirming the role of DMRT1 in initiating the male pathway. These results are shown in the simulation (KO – LOF, E(i) – GOF (i))

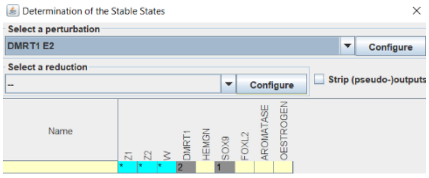

Determination of the Stable States

Select a perturbation: DMRT1 KO

Select a reduction: --

☐ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*				1	1	1

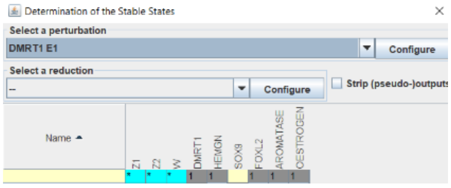

Determination of the Stable States

Select a perturbation: DMRT1 E2

Select a reduction: --

☐ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*	2		1			


Determination of the Stable States

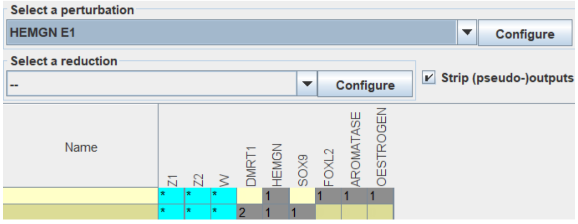
Select a perturbation: DMRT1 E1

Select a reduction: --

☐ Strip (pseudo-)outputs

Name ▲	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*	1	1		1	1	1

Role of HEMGN in Upregulating DMRT1:
HEMGN is essential for upregulating DMRT1, which is crucial for maintaining SOX9 expression and ensuring testis development. This is shown in Table 2 where a **HEMGN GOF (E1) mutation in ZW gonads** results in a testis phenotype, highlighting its role in promoting testis development by upregulating DMRT1, as reflected in the second raw of the GINsim model:

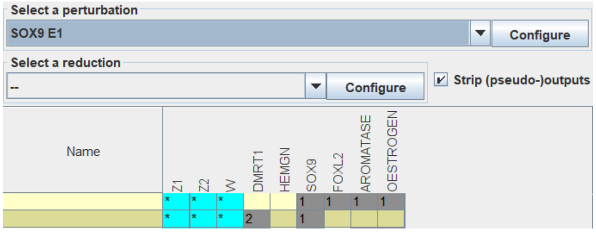

Select a perturbation: HEMGN E1

Select a reduction: --

☑ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*		1		1	1	1
	*	*	*	2	1	1			

SOX9 Autoregulation as a Determinant for Male Development:
SOX9 autoregulation is a key factor for male development. Table 2 shows that a **SOX9 GOF (E1) mutation in ZW gonads** results in a testis phenotype, underscoring the importance of SOX9 in driving the male developmental pathway, as reflected in the second raw of the GINsim model:

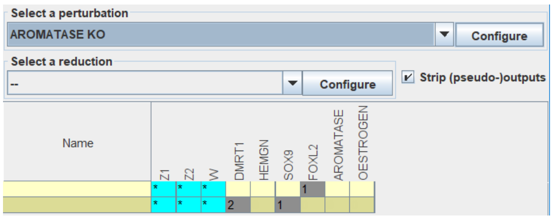

Select a perturbation: SOX9 E1

Select a reduction: --

☑ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*			1	1	1	1
	*	*	*	2		1			

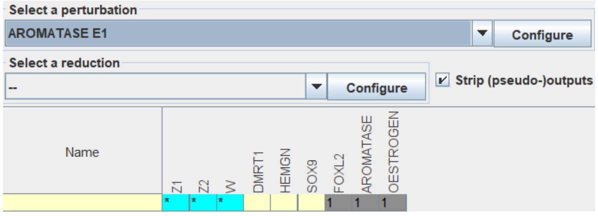
Role of Aromatase in Ovary Development:
Aromatase, the enzyme responsible for converting androgens to estrogens, plays a pivotal role in ovary development by supporting FOXL2 function. In Table 2, an **aromatase LOF (KO) mutation in ZW gonads** leads to a testis phenotype, demonstrating that the absence of aromatase disrupts normal ovary development and leads to male pathway activation. Conversely, an **aromatase GOF (E1) mutation in ZZ gonads** results in an ovary phenotype, indicating that increased aromatase activity can promote ovary development even in a genetically male (ZZ) background.


Select a perturbation: AROMATASE KO

Select a reduction: --

☑ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*				1		
	*	*	*	2		1			


Select a perturbation: AROMATASE E1

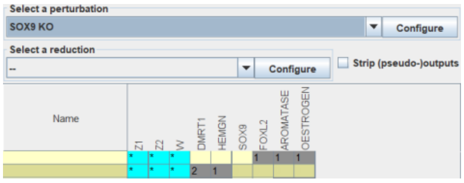
Select a reduction: --

☑ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*				1	1	1

Undecided Gonad Phenotype:

As mentioned before, the authors defined sexual phenotypes based on gene expression: testis identity is indicated by DMRT1 and SOX9 expression with no FOXL2, AROMATASE, or ESTROGEN; ovary identity is indicated by FOXL2, AROMATASE, and ESTROGEN expression with no DMRT1 or SOX9. In the case of a **SOX9 loss-of-function (LOF) mutation in ZZ gonads**, with update setting PC1 (see table above), none of these two conditions is met, and therefore the gonad considers to be neither testis nor ovary, as reflected in the second raw of the GINsim model. This is a significant result, since it sheds light on a situation where the gender of the chicken may be undefined.


Select a perturbation: SOX9 KO

Select a reduction: --

☐ Strip (pseudo-)outputs

Name	Z1	Z2	W	DMRT1	HEMGN	SOX9	FOXL2	AROMATASE	OESTROGEN
	*	*	*				1	1	1
	*	*	*	2	1				

In summary, the article offers a comprehensive and detailed exploration of the genetic and developmental processes underlying primary sex determination in chickens. By integrating current hypotheses and providing a Boolean model that can predict the effects of genetic mutations, Sánchez and Chaouiya significantly advance our understanding of avian sex determination, highlighting the complex interplay of genetic factors that drive the development of sexual characteristics.