

The roles of noise in development

Lessons from *Dictyostelium discoideum*

Daniel Kornai BIOL0044 Project

Supervisor: Professor Christopher Thompson

Abstract

In biological systems, noise is defined as phenotypic variability among cells (or organisms) that are non-genetic in their origin. Multicellular development requires precise coordination of spatiotemporal gene expression patterns across populations. How can robust and deterministic developmental programs emerge from noisy systems?

*Due to its experimental tractability, the social amoeba *Dictyostelium discoideum* has become a leading model organism in the research of noise during development. This paper explores two distinct modes of handling noise in the context of *Dictyostelium*.*

*During deterministic cell fate choices that affect a population of cells in an identical manner, noise must be buffered to maintain the robustness of outcomes despite cell-cell variations. The coordination of developmental initiation typifies the use of buffering. Before development, *Dictyostelium* cells are exposed to varying environmental cues during the independent unicellular part of the lifecycle, which lead to inherent variability within the population. To overcome this variation, cell-cell communication with signalling molecules is coupled to positive feedback loops, which enables the robust collective acquisition of cell fates.*

During stochastic cell fate choices when homogenous populations must generate diverging cell fates, noise is amplified, driving the differentiation of cell types. The acquisition of prestalk and prespore cell fates via lineage priming in later development typifies this mechanism. Heterogeneities in physiological parameters and cell cycle phase help set differing response thresholds to morphogens, allowing cells exposed to identical signals to break symmetry. Following this initial step, lateral inhibition can be used to reach the optimal ratio of cell types.

Introduction

Noise and stochasticity in development

What is noise? Where does it come from?

It is a classical axiom of biology that *phenotype = genotype + environment*. This clearly implies that genetically identical organisms in homogenous environments will exhibit identical phenotypes. However, this simplistic notion has been continually disproved. For example, in clonal *B. subtilis* colonies grown in homogenous environments, approximately 20% cells will be competent (a state where growth ceases and cells gain the ability to take up environmental DNA) while the remaining 80% will grow normally (Losick & Desplan 2008) (**Fig. I1**). Such observations imply that the manifestation of phenotype from genotype is not a fully deterministic process. As such, for the purposes of this text, *noise is defined as phenotypic variability among biological entities (cells, organisms) that are non-genetic in their origin*.

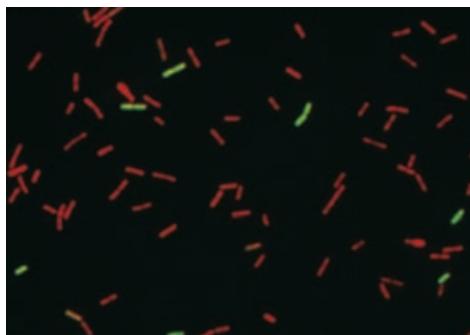


Figure I1: Stochastic distribution of competence status in *B. subtilis*. Cells were visualised using a red strain, GPF was under control of the master competence regulator ComK. Source: Losick & Desplan 2008

This non-genetic variability can be decomposed into intrinsic and extrinsic noise, depending on the mechanisms by which it is generated (Elowitz et al 2002). *Extrinsic noise* is variability that arises from cell-cell differences in cell-specific factors such as cell size, cell cycle position, nutritional status, or the level of various signalling molecules (Singh & Soltani 2013). On the other hand, *Intrinsic noise* is the product of unavoidable stochasticity in biochemical reactions that utilize a finite number of molecules. Due to the low copy number of most genes (1-2 per cell), the process of gene expression is especially stochastic (Eldar & Elowitz 2010). Molecular features at all stages of expression, such as the type and number of transcription factor binding sites, epigenetic modifications such as methylation, and even the ribosomal binding site during translation have been shown to impact the extent of random fluctuations in mRNA and protein level (see Eling et al 2019 for a comprehensive review).

The continued exposure to sources of stochasticity, and the fact that fluctuations in mRNA and protein levels affect the function of downstream targets within the gene regulatory network mean that cells can accumulate significant amounts of non-genetic heterogeneity throughout their lifetime (Dong & Liu 2017) (**Fig. I2**)

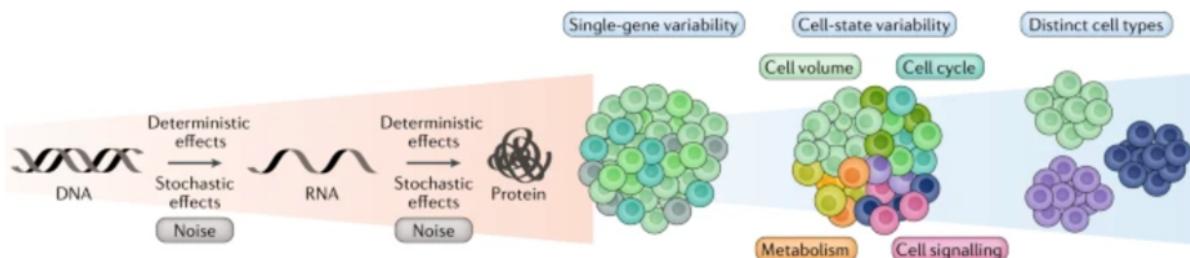


Figure I2: The process of manifesting phenotypes from genotypes is inherently vulnerable to the effects of intrinsic noise (red), and extrinsic noise (blue), leading to the emergence of significant cell-cell heterogeneity among otherwise identical cells. Source: Eling et al 2019

Why is noise important in development and developmental research?

Developmental processes in multicellular organisms are orchestrated, mediated, and manifested by precise changes in spatiotemporal gene expression patterns, which in turn drive the proliferation, lineage commitment, migration, and signalling decisions of individual cells, which over time leads to large scale morphogenesis and functional specification (Dong & Liu 2017).

As introduced previously, the molecular mechanisms responsible for manifesting these genetic programs are highly stochastic. Additionally, theoretical studies show that the buffering of noise requires exceedingly high concentrations of molecules and energy expenditure, and that even the most optimal systems can only reduce noise with the fourth root of the molecules involved (Balazsi et al 2011, Viney & Reece 2013). If noise is a fundamental property of biological systems, and is costly to reduce, how can this be reconciled with the observation of complex, reproducible, and robust developmental programmes that are executed using these noisy systems?

To effectively answer these questions related to the role of noise in development, distinguishing random variation from consequential heterogeneity is crucial. As such, Altschuler & Wu (2010) suggest that the precise measurement and manipulation of heterogeneity in cell populations is key to the effective research of noise.

***Dictyostelium discoideum* as a model organism**

Why is *Dictyostelium* ideal for studying noise in development?

In terms of a multicellular developmental process that can be manipulated and measured at the single cell level, few model organisms are able to compare to the social amoeba *Dictyostelium discoideum*, and its kin the *Dictyostelia*. *Dictyostelium* is highly amenable to a variety of genetic and experimental approaches. Most known *Dictyostelia* species can be cultured on bacterial lawns or grown as mass cultures in liquid media. This in turn enables the purification of biochemical products and provides an easy target for scRNA sequencing. *Dictyostelium* geneticists have also adapted a variety of techniques to the organism. These include gene modification by Cre-loxP and CRISPR/Cas9, or the expression of genes using inducible or constitutive promoters with a large variety of tags that enable the visualization or isolation of cognate proteins (Schaap 2011, Muramoto et al 2019).

While *Dictyostelium* development (detailed below) generates specialized cell types in complex patterns, the low number of cell types and the short developmental time (24 hours) has enabled unmatched insight into the developmental process (Loomis 2015). Additionally, these processes often involve orthologs from metazoans and humans, facilitating the broad application of results to the field of development (Bozzaro 2019).

As such, *Dictyostelium* has served as a crucial model organism in developmental research, helping to uncover the mechanisms involved in cellular processes such as chemotaxis and cytokinesis. *Dictyostelium* is also a leader in the research of morphogenesis, helping scientists understand how the control of cell movements and regulated cell differentiation is achieved during multicellular development (Schaap 2007). Owing to the ease with which scRNA sequencing can be performed on colonies, *Dictyostelium* has become one of the leading organisms for researching the role of cell-cell heterogeneities in multicellular development. In this text, some of these recent advances in heterogeneity research will be presented, highlighting parallels to developmental mechanisms that have been identified in other organisms

The *Dictyostelium* Lifecycle

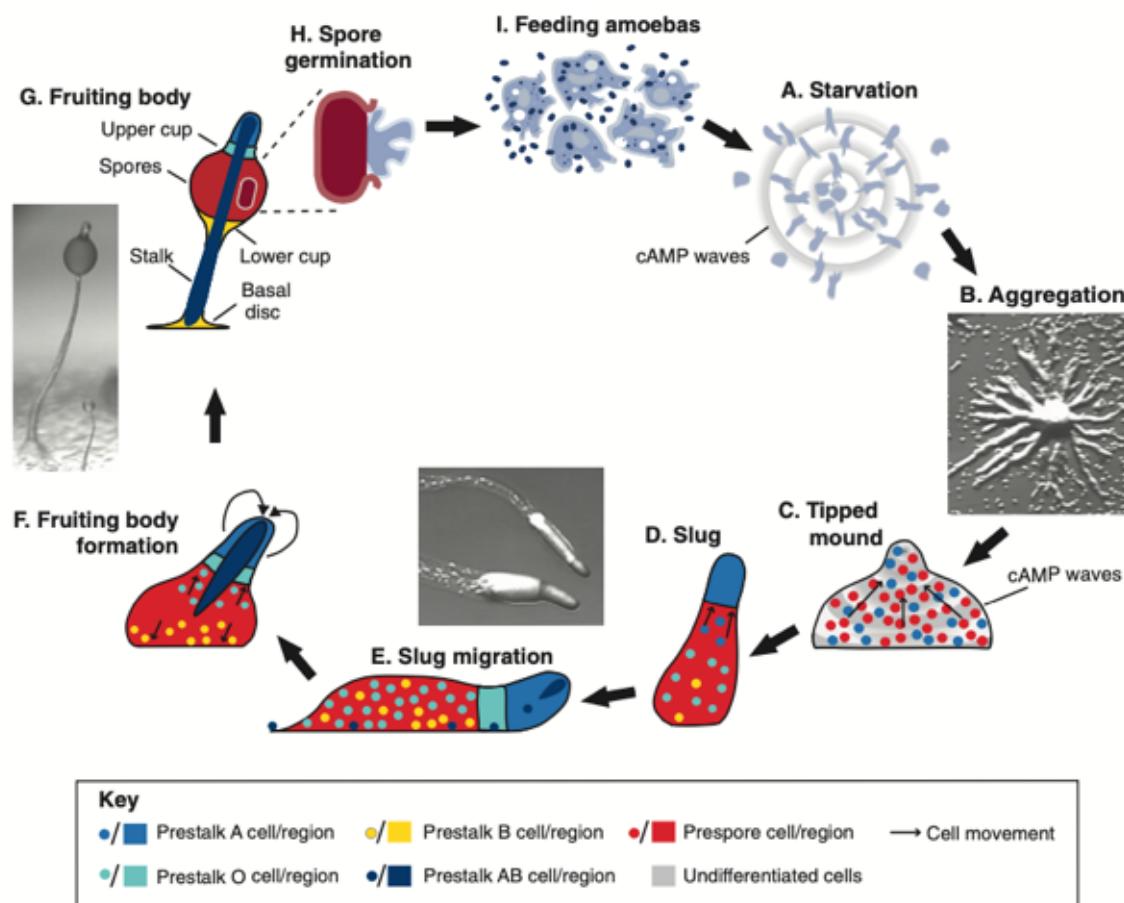


Figure I3: Developmental lifecycle of the social amoeba *Dictyostelium discoideum*. Source: Schaap 2011

In soil habitats around the world, from the artic to tropical regions, and in deserts and rainforests, Dictyostelid social amoebas exist as solitary unicellular organisms predating bacteria (Swanson et al 1999) (Fig. I3-I). As starvation nears, *Dictyostelia* begin to prepare for the social-developmental lifecycle by secreting intercellular signals. The glycoprotein PSF is secreted at a constant rate during growth, allowing cells to estimate the local density of kin. As starvation becomes imminent, a second glycoprotein, CMF is also secreted into the environment (Consalvo et al 2019).

When cells detect large concentrations of PSF, CMF, and do not detect bacteria, a variety of transcriptional changes occur that enable the synthesis and detection of cAMP (Loomis 2014). Among the newly transcribed genes is the G-protein coupled cAMP receptor CAR1, adenylate cyclase A (ACA) which synthesizes cAMP, and the extracellular cAMP degrading phosphodiesterase PdsA.

The production, sensing, and extracellular degradation of cAMP is eventually coordinated to enable the generation of cAMP pulses (Schaap 2011) (**Fig. I3-A**). These periodic waves coordinate the movement of 10^2 to 10^6+ cells, allowing them to stream and aggregate into the multicellular “mound” (Schaap 2016) (**Fig. I3-B**).

Inside the mound, cells begin to differentiate into two cell types. The emerging prestalk (PST) and prespore (PSP) cells initially appear in random locations within the mound (Thompson et al 2004). However, PST cells prefer to chemotax towards cAMP, while PSP cells prefer to stick to PST cells, resulting in the gradual separation of cell types, with PST cells at the tip of the mound, and PSP cells at the bottom (Siebert & Weijer 1995, Fujimori et al 2019) (**Fig. I3-C**). PST cells at the tip of the mound continue to emit cAMP pulses. The resulting chemotaxis pushes the mound upwards, causing the emerging column of cells to tip over, forming the “slug” (Nicol et al 1999) (**Fig. I3-D**).

As such, PST cells are at the anterior of the slug, while PSP cells are in the posterior. Following chemotactic and phototactic signals, the slug migrates towards the top layer of soil. Slugs protect cells from predators, allow cells to tolerate noxious environments, and enable the migration of large distances to locations where new food is available (Trenchard 2019) (**Fig. I3-E**). During migration, the 30:70 ratio of PST to PSP cells is stabilized using lateral inhibition. In this system, PSP cells secrete the chlorinated cyclohexanone DIF-1, which induces the differentiation of PST cells (specifically PST-O, and PST-B subtypes), and blocks the differentiation of further PSP cells. PST cells secrete cAMP, which induces the differentiation of PSP cells, and blocks the differentiation of further PST cells. When PST:PSP ratio deviates from the optimum, the imbalance between the cell type inducing signals will induce the transdifferentiation of cells until the optimal ratio is reached (**Fig. I4**) (Pineda et al 2015, Mohri et al 2020).

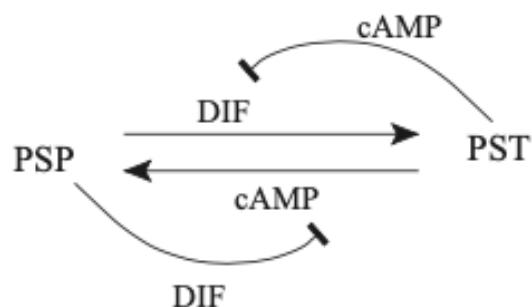


Figure I4: Lateral inhibition of cell types via DIF and cAMP. Source: Pineda et al 2015

When the movement of the slug is arrested, it begins to form the fruiting structure (“sorocarp”) (**Fig. I3-F**). This involves the terminal differentiation of cells. PST-B cells form the basal disc, which supports the tapered stalk (made of PST-A cells), which uses a lower cup formed of PST-B cells, and an upper cup formed from PST-O cells to hold a spherical mass of PSP derived spore cells aloft (Schaap 2016) (**Fig. I3-G**). The spore cells synthesize a multi-layered spore wall, which facilitates survival in unfavorable environments until they germinate into unicellular amoeba to begin the cycle again (**Fig. I3-H**) (Schaap & Schilde 2018).

The perspective of this text

Which aspects of noise will be considered?

As the above section has described, the development of *Dictyostelium* is immensely complex. How can the perspective of noise help to better understand the function of the various systems involved? In their review of the role of transcriptional noise in cell fate specification, Urban & Johnston (2018) establish that noise is handled in two distinct modes depending on the goals of the given developmental stage. When executing deterministic developmental decisions that are meant to affect a population or subpopulation of cells in an identical manner, noise can impair the development process, and the effects of noise must be buffered to ensure the reproducibility of outcomes despite cell-cell variations. In contrast, during stochastic developmental decisions when seemingly homogenous populations must generate diverging cell states in subpopulations without external control, sources of noise are relayed and amplified to generate diversity in cell fates. In my opinion, *Dictyostelium* development shows clear and experimentally validated examples of both distinct strategies, making it a prime choice for investigating the role of noise in development.

The content of this paper

Section 1 will explore how the buffering of noise and stochasticity enables the deterministic cell fate choice to enter development. The initiation of the social cycle in *Dictyostelium* is ideal for studying buffering, because independently growing unicellular amoeba are exposed to a large range of microenvironments, generating significant variations in starting cell states. To achieve the “lock-step synchrony” (Rosengarten et al 2015) that characterizes later development, this initial variation must be significantly filtered during the initiation stage. To gain a rounded perspective on the buffering of noise during *Dictyostelium* development initiation, the phenomena will be investigated from multiple perspectives.

First, a comparative phylogenetic approach will be utilized to explore the unique properties of group 4 *Dictyostelium* aggregation compared to more basal group 1-3 *Dictyostelia*. These analyses reveal that group 4 innovated the use of cAMP signalling during early development, and that group 4 multicellularity involves significantly larger numbers of cells and increased specialization.

Because these results suggest that early developmental signalling in group 4 is more effective at synchronizing larger populations, and thus more effective at filtering noise, the biochemical mechanisms involved in coordinated entry into early development will be investigated. This will reveal that the growth-differentiation pathway (GDP) incorporates a positive feedback loop between cAMP production and sensing, a type of feedback loop that has been implicated in the stable induction of cell fates in heterogenous environments such as cancer.

Following this, the information gained in previous sections will be incorporated into an original model of GDP controlled developmental initiation in a population. This model will be utilized to explore how extrinsic and intrinsic sources of noise impair synchronized development, and why the positive feedback loop architecture in *Dictyostelium discoideum* enables it to overcome the difficulties caused by noise.

At the end of the first section, overarching conclusions will be presented about the role of stochasticity buffering during development.

Section 2 will explore how noise can drive the acquisition of diverging cell fates from an otherwise homogenous population during the mound stage. Specifically, I will investigate how *Dictyostelium* is able to break symmetry by utilizing multiple sources of cell-cell heterogeneity and generate consistent ratios of specialized cell types across a variety of environmental conditions and population sizes by using lateral inhibition to correct mistakes in the proportion of cell types.

First, the developmental concepts of lineage priming, and salt-and-pepper differentiation will be introduced, along with experimental evidence that *Dictyostelium* development utilizes these mechanisms, which together suggest that symmetry breaking in *Dictyostelium* utilizes stochastic processes.

Following this, the sources of cell-cell heterogeneity will be explored, highlighting how cell parameters such Ca^{2+} levels, and cell cycle stage biases cells towards either the prestalk or prespore fate.

After this, some recent work of the Thompson lab and their collaborators will be introduced, which has begun to elucidate how *Dictyostelium* can utilize cell cycle desynchronization in populations to induce robust differentiation, and unpublished results which demonstrate that epigenetic modifications can act as a backup source of stochasticity when cell-cycle dynamics are grossly disturbed.

Finally, I will introduce how lateral inhibition can be utilized to correct imbalances in the proportion of cell types, highlighting the crucial role of DIF and cAMP signalling.

Rounding out section 2, general conclusions about the mechanisms which harness stochasticity will be presented.

Finally, In the **Future research** section, a general outlook on the field of stochasticity in development will be interspersed with concrete recommendations for experiments in *Dictyostelium* and beyond.

Section 1:

The buffering of noise during developmental initiation

Introduction

Dictyostelium development is initiated in the absence of food (Schaap 2016). While bacteria are plentiful, cells exist as solitary unicellular organisms. Because cells make decisions independently, localized variations in the density of their bacterial food source, and various physical conditions such as osmolarity or temperature act as significant sources of extrinsic noise at the population level. Additionally, intrinsic noise from the stochastic dynamics of gene expression also leads to the further divergence of cell states. If environmental variation and cell-cell variation are the status quo during the pre-development phase of the lifecycle, how can *Dictyostelium* guarantee that all cells choose to enter the social-developmental cycle at identical times? In Section 1, I will attempt to provide a comprehensive answer to this question.

Section 1A: The phylogeny of Dictyostelia

Dictyostelid social amoebas participate in sorocarpic (fruiting body) multicellularity, a form of cooperative strategy that has evolved independently at least 8 times. Using genetic data, all currently known species of social amoebas can be classified into 4 major monophyletic groups (**Fig. 1A-1**) (Schaap 2011, Schaap 2016).

Species in group 1-3 utilize glorin or folic acid for chemoattraction, form small, clustered fruiting bodies that lack supporting discs, and form stalk cells during culmination using the positional redifferentiation of prespore cells. These species only utilize cAMP at the last stages of development when inducing spore maturation.

Dictyostelium Discoideum belongs to Group 4, along with other species that utilize cAMP to coordinate aggregation, have a well-defined proportion of prestalk and prespore cells at the slug stage, and build large robust solitary fruiting bodies from more numerous and larger spores, that are also supported by basal discs (Schaap 2007, Schilde et al 2014, Schaap 2016, Kawabe et al 2019).

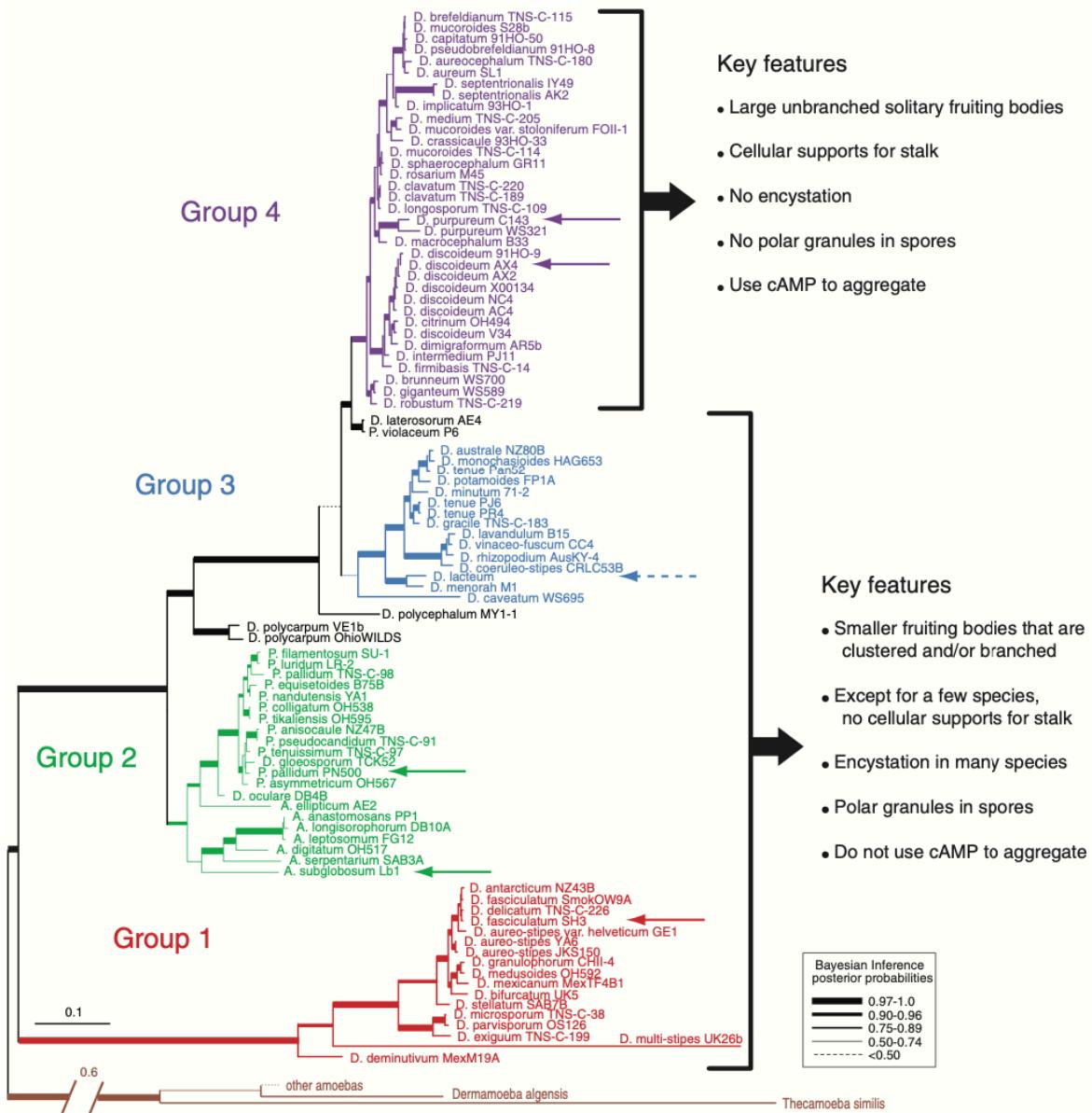


Figure 1A-1: Dictyostelia Phylogeny constructed using Bayesian inference. Source: Schaap 2011

Section 1B: Function and architecture of the buffering mechanism

Larger groups require more extensive buffering

Comparative analysis has highlighted that the use of cAMP signaling during early development is a unique innovation in group 4 Dictyostelia, and that group 4 multicellularity involves a significantly higher number of cells than in groups 1-3 (Schaap 2016). In my opinion, larger groups of cells contain more stochasticity than smaller groups. This is because:

- 1) higher number of cells inhabit larger areas, which implies that cells are subjected to a wider range of microenvironments. For example, bacterial concentrations are more likely to be distributed such that one subpopulation is starving while others are not. This leads to more extrinsic noise.
- 2) A higher number of cells are distributed across a larger area, meaning that any cell-cell communication that relies on the diffusion of signaling molecules is less efficient at transferring information between the most distant cells.

As larger groups contain more stochastic variation, this implies that systems able to synchronize the behavior of larger groups can buffer more stochasticity. From this, the question arises: What properties of the group 4 initiation signalling system make it so efficient at overcoming noise? To answer this, I investigated the signalling system controlling the transition into development.

The biochemical and genetic mechanisms of buffering

Dictyostelium cells continuously secrete the PSF (Prestarvation Factor) into the growth medium. As such, PSF concentrations increase as cell density increases, allowing cells to estimate the density of kin in their surroundings (Consalvo et al 2019). Crucially, PSF response is modulated by the presence of bacteria, such that low nutrient availability increases sensitivity to PSF. Combined, this means that when the ratio of nutrient availability and cell density crosses a threshold where cells expect to starve soon, they can begin to prepare for later development.

Inside starving cells that are exposed to PSF, the expression level of the protein kinase YakA rises. YakA in turn inhibits PufA, which is a translational inhibitor of the amp dependent protein kinase PKA. By inhibiting the inhibitor of PKA, PSF indirectly activates PKA. The activity of PKA then leads to the accumulation of the cAMP cell surface receptor CAR1, and the adenylyl cyclase ACA, which converts ATP into cAMP (**Fig. 1B-2**). Simultaneously, CMF (Conditioned Medium Factor) is also produced by the growing cells. While cells begin to accumulate CMF before starvation, the export of CMF, and the translocation of CMF receptors to the cell surface is only initiated after starvation. After cells have been exposed to sufficient levels of CMF, the binding of cAMP to CAR1 will stimulate ACA to make more cAMP, which is then secreted into the environment, completing a positive feedback loop (**Fig. 1B-2**) (Jain & Gomer 1994, Loomis 2014).

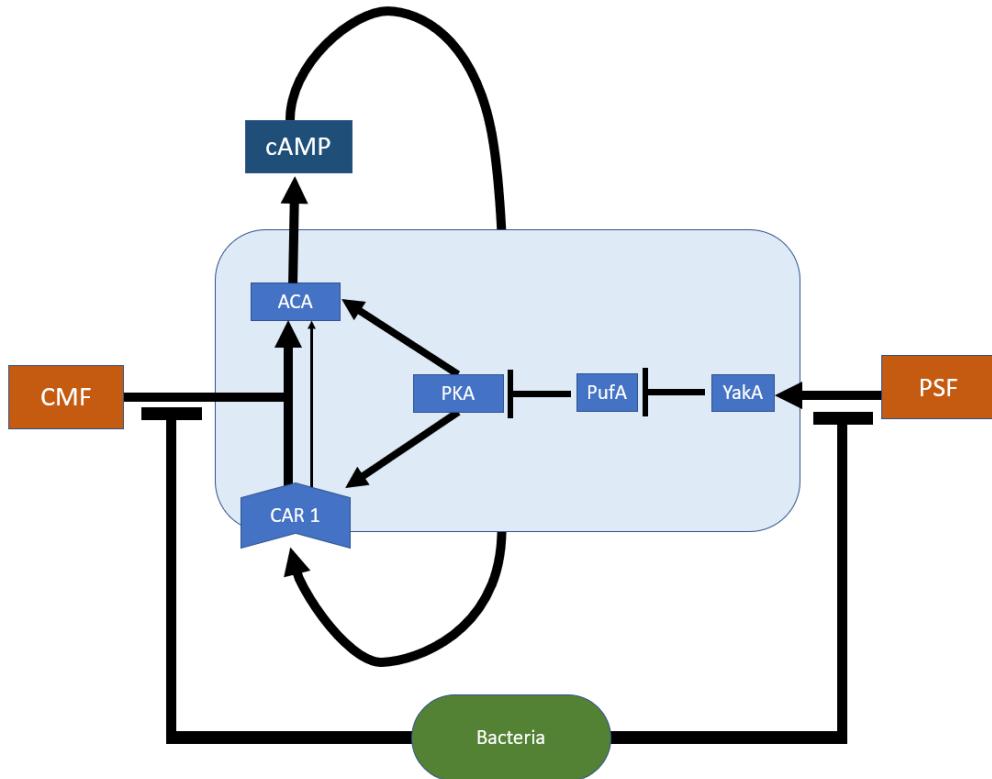


Figure 1B-3: Simplified view of the aggregation competence pathway. Growing cells secrete PSF and CMF into their environment, which increases the level of these signals as density grows. When bacterial density is low, PSF can activate YakA, which inhibits PufA, the inhibitor of PKA. PKA activity leads to the activation of CAR1 and ACA, which form the components of the positive feedback loop responding to cAMP signals. Simultaneously, exposure to CMF allows CAR1 to stimulate ACA, completing the feedback loop.

Due to the positive feedback loop, cAMP concentrations within the soil will begin to rise throughout the entire colony. Exposure to cAMP is crucial for the expression of many genes involved in later development, such as the cell adhesion genes *csA*, *tgrB1*, and *tgrC1* (Schaap 2016). Thus, cAMP in early development acts to synchronize the transcriptomes of cells across the population.

Positive feedback loops drive cell fate decisions in noisy environments

The use of positive feedback loops to drive the coordinated entry of a population into a new cell fate is not unique to Dictyostelia. For example, cancers utilize positive feedback loops to achieve robustness, and guide the progression into metastatic states (Radisavljevic 2012). Basal-like breast cancer is associated poor prognosis and with high grade distant metastasis. In this type of cancer, the transcription of AKR1B1 by Twist2 results in the activation of nuclear factor factor κ B (NF- κ B). NF- κ B in turn upregulates Twist2 expression, inducing a positive feedback loop that guides the activity of the epithelial-mesenchymal transition program, leading to increased tumorigenicity (Wu et al 2017).

Section 1C: Modelling the growth-differentiation pathway

What are the goals of the model?

Phylogenetic analysis found that group 4 species have significantly larger aggregate sizes than groups 1-3, which implies an increased capacity to overcome noise at the initiation of development. A closer examination of the growth-differentiation pathway (GDP) in *Dictyostelium discoideum* has revealed that cAMP production and sensing are coupled in a positive feedback loop, a motif that has been suggested to drive robust collective cell fate decisions in cancer. Combining the above, the question arises: Did the coupling of cell-cell signalling to a positive feedback loop enable group 4 Dictyostelia to lower the amount of stochastic variation in the onset of development, thereby allowing larger groups to aggregate successfully?

A model for investigating this question needs to have three important attributes. First, it must be biologically realistic, reproducing the activity of the GDP using the concentration of signalling molecules as both input and output parameters and incorporating the interactions between the most important genes. Second, it needs to implement various sources of noise as controllable parameters. Third, a model simulating a single cell must be extendable to multiple cells, and the level of intercellular cAMP signalling should be a controllable parameter.

In the following sections, I will start by introducing Boolean networks for modelling gene networks. Next, I will demonstrate how the GDP can be transformed into a Boolean network. Finally, I will extend the model to multiple cells, and investigate how noise impairs the ability of populations to synchronize, and how cell-cell communication coupled with positive feedback loops can overcome these difficulties.

Modelling biochemical networks with Boolean logic

Because the model is meant to investigate population level behaviour rather than accurately simulate intracellular processes, Boolean gene networks utilizing two possible internal states for each gene emerged as the primary approach (**Box 1**). However, the growth-differentiation pathway relies on gradual changes in the level of signalling molecules over time, meaning that Boolean circuitry must interact with continuous concentrations of signalling molecules (**Box 2**).

Box 1: Primer on Boolean gene networks and Boolean circuits

Boolean gene networks.

Boolean gene networks are a model of real gene networks which consist of variables representing genes which have a state of either 0 or 1 (corresponding to gene activity being above or below of some threshold required for its downstream effects), and a set of Boolean functions which determine the state of a given gene using the state of its regulators as inputs (Giacomantonio & Goodhill 2010). While two states per gene may appear to be simplistic, Boolean logic can effectively characterize the activity of gene networks when the transcriptional response between regulator and target is definitive and fast (McAdams & Shapiro 1995, Shmulevich et al 2002).

Representing Boolean logic as circuits.

Instead of thinking about Boolean networks in their native mathematical form of equations, they can be represented as an electronic circuit, which makes the interactions within the network easier to understand. In these circuits, Boolean operations are performed using logic gates. Logic gates take one or more binary inputs and produce a single binary output. In the current model, only three types of gates will be used. For each type, a truth table illustrating the output of a given gate for every possible combination of inputs. and visual symbol will be provided (Fig. B1).

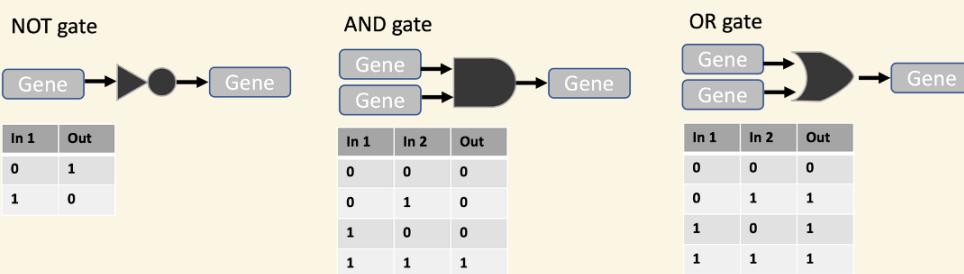


Figure B1: Truth tables of various gates. Left: The NOT gate takes one input, and outputs the opposite of the input state. Middle: The AND gate takes two inputs, and only outputs 1 if both inputs are 1. Right: The OR gate also takes two inputs, and outputs 1 if one or both inputs are 1.

Box 2: Bridging Boolean networks and continuous concentrations

While Boolean gene networks can be a good proxy for the internal state of the cells, the activity of network components in the growth-differentiation pathway is dependent on gradual changes in signalling molecule concentrations, which are non-Boolean. Because the gene network is simulated over time, the concentration of signalling molecules can be transformed into the probability of receiving or emitting a binary signal using two interfacing elements. The signal molecule sensor (SMS) takes the concentration of the signalling molecule c as an input. In each time step, the sensor has a $c/100$ probability of activating its target gene (Fig. B2 left). The signal secretor (SS) takes the activity state of its controlling gene as input. For each time step where the control gene is on, the signal emitter releases p amount of signalling molecule (Fig. B2 right). Using these two interfaces, Boolean networks can be adapted to utilize continuous concentrations of signalling molecules as both inputs and outputs



Figure B2: Left: Signalling molecule sensors are responsible for converting the concentrations of signalling molecules into the probabilistic binary activation of genes. Right: Signal secretors are responsible for converting the binary activity of signalling molecule production genes over time into changes of concentrations.

Identifying sub-pathways in the growth-differentiation network

Because the GDP includes multiple inputs and components, translating the network into a Boolean circuit is difficult. To make the model and modelling process more transparent, the GDP was split into 3 sub-pathways (**Fig. 1C-3**).

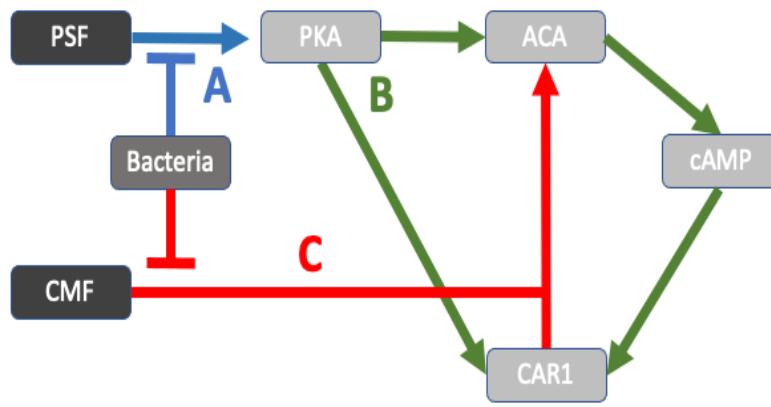


Figure 1C-3: Outline of the simplified growth-development pathway. Pathway A (blue) represents the bacterially inhibited ability of PSF to activate PKA. Pathway B (green) represents the PKA mediated induction of cAMP sensing capability, and cAMP production capability. Pathway C (red) represents the capability of cells exposed to CMF to couple cAMP sensing to cAMP production, facilitating the emergence of a positive feedback loop.

Sub-pathway **A** represents the bacterially inhibited PSF mediated activation of PKA (**Fig. 1C-4 left**). While the actual regulatory network works because PSF inhibits the inhibitor of PKA, this represents a redundant double negation in Boolean logic and is thus simplified such that PSF directly activates PKA. Integrating the effect of bacteria into this reduced network, PKA should only become active when PSF is 1, and Bacteria is 0 (**Fig. 1C-4 middle**). To produce this behaviour, the input from the Bacterial SMS is passed through a NOT gate, after which it is combined with the PSF SS input using the AND gate G1 (**Fig. 1C-4 right**).

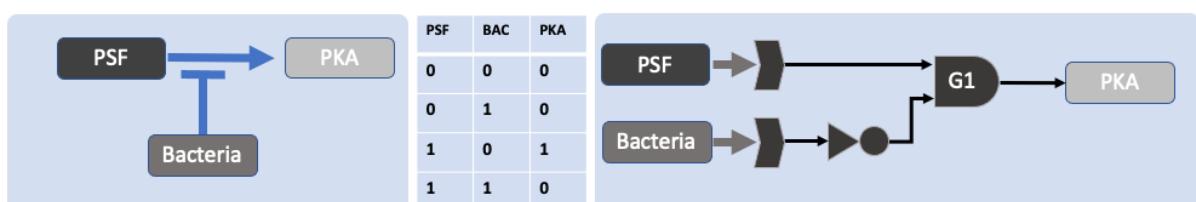


Figure 1C-4: Sub-pathway A. Left: Visual representation of the primary mechanisms involved in PSF signal transduction. Middle: Truth table describing the input and output states of the sub-pathway. PSF can only activate PKA when no bacteria are present. Right: Diagram of electronic circuit that produces the input and output relations specified in the truth table.

Sub-pathway **B** represents the activation of ACA and CAR1 by PKA, the production of cAMP by ACA, and the activation of CAR1 by cAMP (**Fig. 1C-5 left**). While this sub-pathway does not apply any Boolean function, the production, accumulation, and sensing of cAMP is the central mechanism in the growth-development pathway. Inside the circuit, the state of ACA and CAR1 is linked to PKA such that PKA activates CAR1 and ACA immediately.

When ACA becomes activated, a SS releases p amount of cAMP into the environment. The concentration of extracellular cAMP c acts as the input to a SMS, such that CAR1 is activated with a chance proportional to c . As such, the activity of the cAMP feedback loop is not immediate, only becoming continuous when cAMP concentrations are sufficiently high.

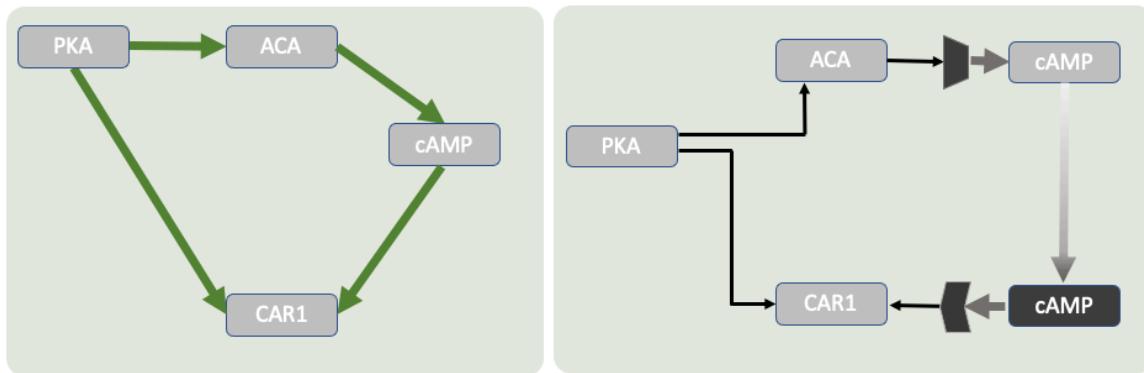


Figure 1C-5: Sub-pathway B. Left: Visual representation of gene network activity. PKA activates ACA, causing the production of cAMP, and CAR1, allowing cells to sense cAMP. Right: Diagram of electronic circuit reproducing pathway behaviour. The accumulation of cAMP in the environment due to ACA activity is modelled using a signal secretor. The density dependent activation of CAR1 by cAMP is modelled using a signalling molecule sensor.

Sub-pathway **C** represents the fact that populations that are unable to accumulate CMF are also unable to efficiently produce cAMP. This is because CMF is required for the G protein mediated CAR1 stimulation of ACA. Additionally, CMF receptors only activate at low bacterial density (Jain & Gomer 1994) (**Fig. 1C-6 left**). Transforming these biological observations into a truth table, we find that ACA should only output 1 if CMF is 1, BAC is 0, and CAR1 is 1 (**Fig. 1C-6 middle**). To reproduce the bacterial inhibition of CMF signalling, we reuse the design of **A** (**Fig. 1C-6 right**). Laboratory experiments also demonstrated that brief (10s) exposure to CMF is enough to permanently enable CAR1 mediated ACA stimulation. As such, an additional OR gate (G3) was added between G2 and G4. Because the output of G3 is looped back into its own input, once a signal passes through it, it will become permanently enabled, allowing CAR1 to stimulate ACA without further CMF signals (**Fig. 1C-6 right**).

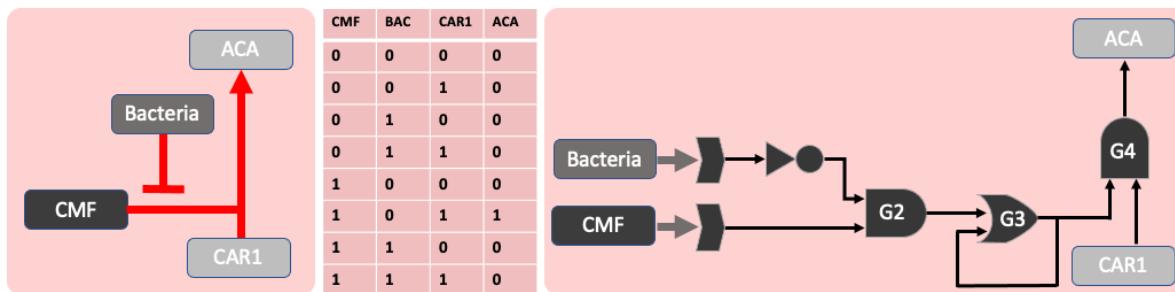


Figure M6: Sub-pathway C. Right, visual representation of gene network activity. CMF can enable CAR1 to stimulate ACA when no bacteria are present. This allows the sensing of cAMP to stimulate the production of more cAMP, creating a positive feedback loop. Middle: Truth table representing the idealized behaviour of the sub-pathway. CAR1 can activate ACA only when high levels of CMF and no Bacteria are present. Right: Diagram of electronic circuit that produces the input output relations of the truth table. Bacteria and CMF levels are detected using signalling molecule sensors. The OR gate G3 acts as a permanent memory of CMF detection.

Integrating and implementing the model

The final model of a single cell is a combination of A, B, and C (**Fig. 1C-7**). Inputs to the model are the concentrations of PSF, CMF, Bacteria, and cAMP, which affect the internal states of PKA, CAR1, and ACA, which in turn affects the levels of cAMP production. The model was implemented in R as a sequential logic circuit (source code is provided in the appendix).

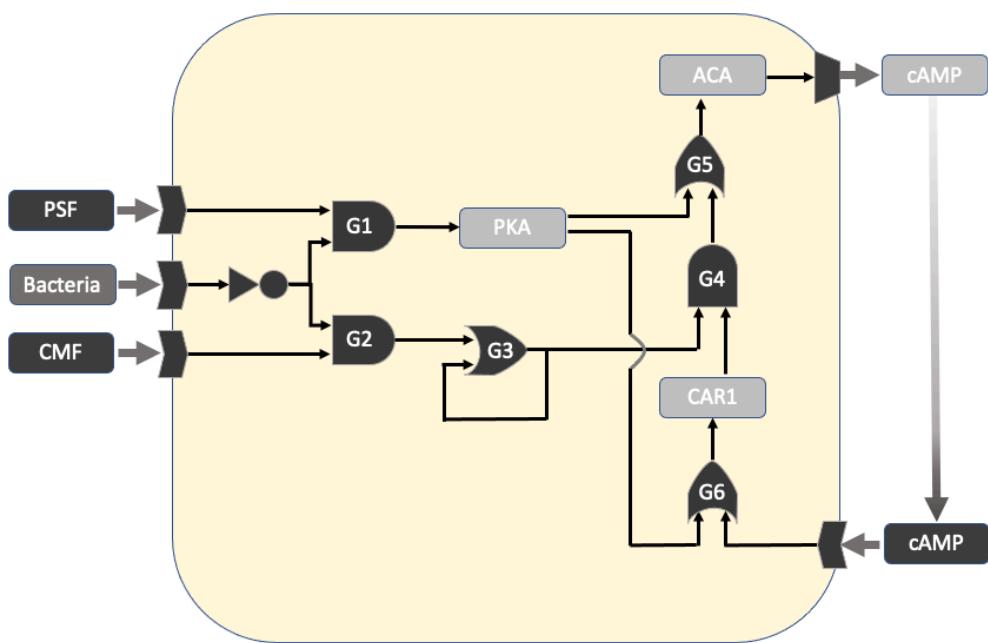


Figure 1C-7: The combined Boolean circuit model of the GDP. The circuit is a combination of A, B, and C, with two additions to account for overlaps: As both PKA and CAR1 could stimulate ACA, OR gate G5 was added, and because both cAMP and PKA could activate CAR1, OR gate G6 was added

To lay the groundwork for cell-cell heterogeneities in signal responsiveness, a signal insensitivity parameter s was implemented, which changes the behaviour of SMSs such that when signal concentration is c , sensors respond with a chance of $c-s/100$ in each time step.

Before the simulation is initiated, starting values for PSF, CMF and bacterial concentrations are set. PSF concentrations are set to a low value between 0 and 100 to simulate a starting point where cell densities are low, Bacterial concentrations are set to a high value between 0 and 100 to simulate abundant food before development is induced, and CMF concentrations are set to 0 because this compound is only released when bacterial resources are exhausted. During each time step, the Bacterial density is decreased, while the PSF concentration is increased by a predefined value. When Bacterial levels reach 0, CMF values also begin to increase.

To verify that the model is exhibiting the anticipated changes in PSF, CMF and Bacterial concentrations, capturing the biologically relevant aspects of GDP behaviour, and producing cAMP after starvation, a simulation of a single cell was performed. Charting the level of signalling molecules, and the activity of circuit components over time reveals the expected pattern (**Fig. 1C-8**).

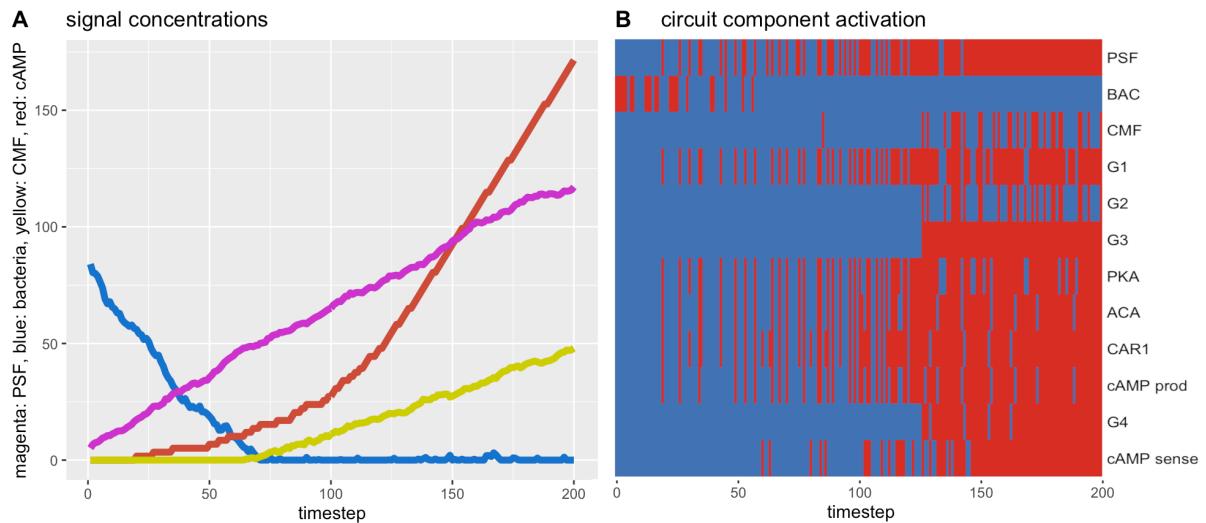


Figure 1C-8: Left: Concentration of signalling molecules, magenta: PSF, yellow: CMF, blue: Bacterial density, red: cAMP. Right: Activation of circuit components over time: blue: inactive, red: active. Observations: As PSF levels rise, its sensor becomes increasingly activated. As bacterial levels decrease, the sensor activates less. When the chance of not receiving a bacterial signal approaches the chance of receiving a PSF signal, PKA becomes more and more active, and cAMP production begins, slowly increasing the concentration of cAMP. When Bacterial levels drop completely, CMF is released, which completes the feedback loop between cAMP production and sensing, allowing cAMP levels to rise at high speed compared to the other signalling compounds.

Extending the model to multiple independent cells

To investigate collective behaviour, the single cell model was extended to a 1-D spatial model of 100 cells. For each cell, the starting value of PSF and Bacterial concentrations was set by randomly drawing from a normal distribution with specified mean and standard deviation values. By raising or lowering the standard deviation of starting values, various degrees of environmental heterogeneity could be simulated. Before the simulation, the signal insensitivity parameter s was also specified for each cell by randomly drawing s values from a normal distribution with a specified mean and standard deviation. By changing the standard deviation of s , various degrees of cell-cell heterogeneity in signal responsiveness could be simulated (**Fig 1C-9**).

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
PSF_conc	12.002960	45.40531	11.80995	4.04014	10.634155	3.339548	1.317313	14.21091	26.36069	26.31293
BAC_conc	91.631353	60.30440	76.65660	88.83696	95.918714	41.264970	59.096672	68.38171	64.53675	109.03270
CMF_conc	0.000000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000
cAMP_conc	0.000000	0.00000	0.00000	0.00000	0.000000	0.000000	0.000000	0.00000	0.00000	0.00000
sig_in sensitivity	1.436646	27.82304	6.06701	11.39393	2.320993	26.852595	33.324254	28.37104	30.70671	14.76979

Figure 1C-9: Subsection of the data table describing the starting states for each cell. PSF and Bacteria levels, and signal insensitivity are highly varied in this population.

When cells export a signalling molecule, it will diffuse throughout the soil and effect neighbouring cells. As such, during each time step, a set proportion (0.05%) of each signalling compound was transferred to neighbouring cells. This results in the very gradual equalization of extracellular signals.

To assess the level of synchronicity of the population, the downstream effects of cAMP signalling on gene expression were modelled. When cells are exposed to extracellular cAMP, the expression of genes required for the aggregation stage is induced (Loomis 2015). For each cell, activity of the cAMP sensor was analysed by calculating the proportion of time it was active within 100 timesteps (this will be plotted as a green line). If cells sensed cAMP in a given timestep, this was considered to increase the transcription level of aggregation specific genes by a small amount (this will be plotted as a purple line). If a cell has transcribed aggregation genes during 100 separate timesteps, it was considered competent for aggregation. The ratio of cell aggregation competent cells was calculated and plotted as a red line. The timestep at which each cell reached competence was also recorded and plotted as a histogram.

Implementation and results of the model

During the first stage of modelling, the goal was to investigate how populations would react to varying degrees of environmental and cell-cell heterogeneity, when the cAMP feedback loop is not coupled to cell-cell communication. This would mean that PSF and CMF would act as intercellular signals, but their downstream effects on cAMP production would be compartmentalized to individual cells. In the model, this was achieved by diffusing PSF and CMF among cells, but not diffusing cAMP, which meant that all cAMP molecules that a cell was exposed to were produced by themselves.

When environmental variables and signal sensitivity are identical, the independent cells will react in a homogenous fashion, accumulating cAMP and transcripts of aggregation genes at similar speeds, leading to a synchronized gain of aggregation competence (**Fig. 1C-10**).

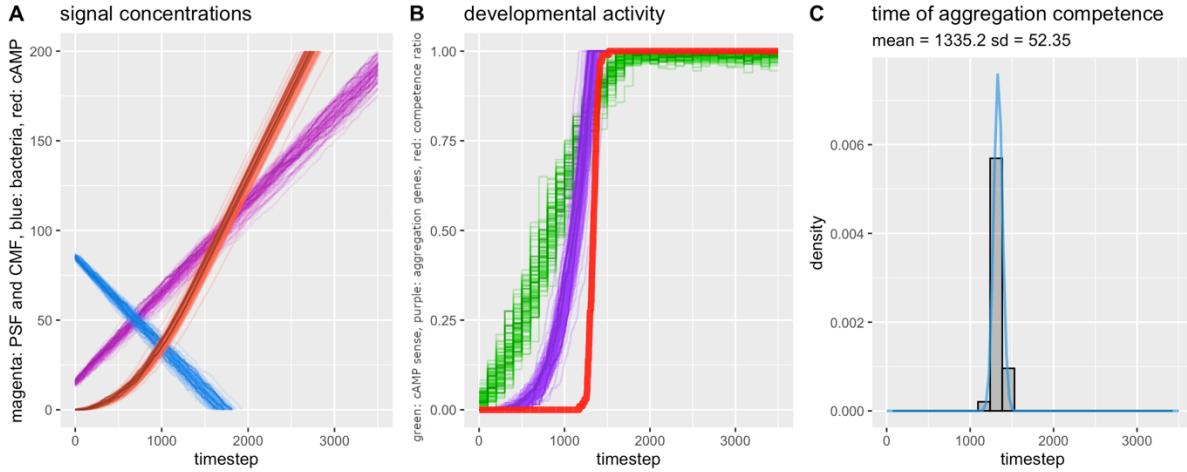


Figure 1C-10: The entry of cells into development is synchronized. A: Concentrations of signalling molecules over time. magenta: CMF, blue: Bacteria, red: cAMP (CMF is not pictured). B: Activity of genetic mechanisms involved in development. green: cAMP sensing activity over 100 timesteps. purple: transcription level of later developmental genes that are upregulated in response to cAMP sensing. red: ratio of cells that have accumulated sufficient levels of later developmental transcripts. C: density plot showing the times when cells reach enough aggregation competence transcripts to gain aggregation competence

The effects of cell-cell heterogeneity on synchronization can be observed by introducing cell-cell variations in signal responsiveness while environmental variables are identical. Despite identical signal cues, cAMP levels around each cell diverge (**Fig. 1C-11A red lines**). Therefore, the activity levels of the cAMP sensors also diverge (**Fig. 1C-11B green lines**), leading to significant variation in the transcription levels of aggregation genes (**Fig. 1C-11B purple lines**), ultimately causing cells to acquire aggregation competence at varying times (**Fig. 1C-11 C**).

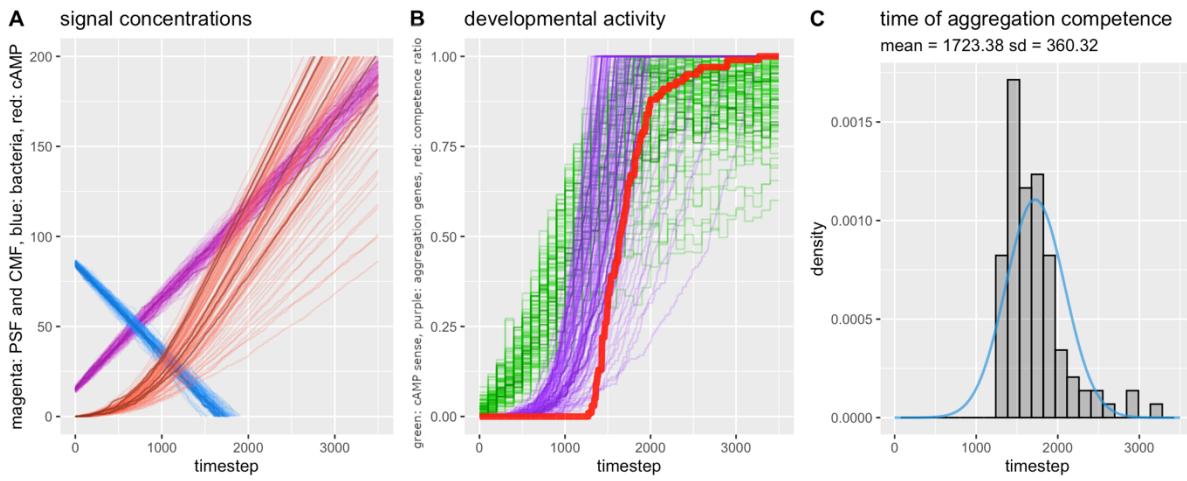


Figure 1C-11: The entry of cells into development is desynchronized by variations in signal sensitivity. Note that certain cells accumulate cAMP at much lower speeds (A: red lines that are significantly below average), and therefore have less activity in the downstream developmental genes (B: purple lines below the average), leading to “late developer” cells which gain aggregation competence far after other cells (C: right hand tail of the distribution)

As the distribution of food sources, and the density of amoeba is not homogenous, significant inter-population variation can exist in the levels of signalling molecules communicating these environmental parameters. Even when signal sensitivity is identical between cells, this will cause divergent entry into development (**Fig 1C-12**).

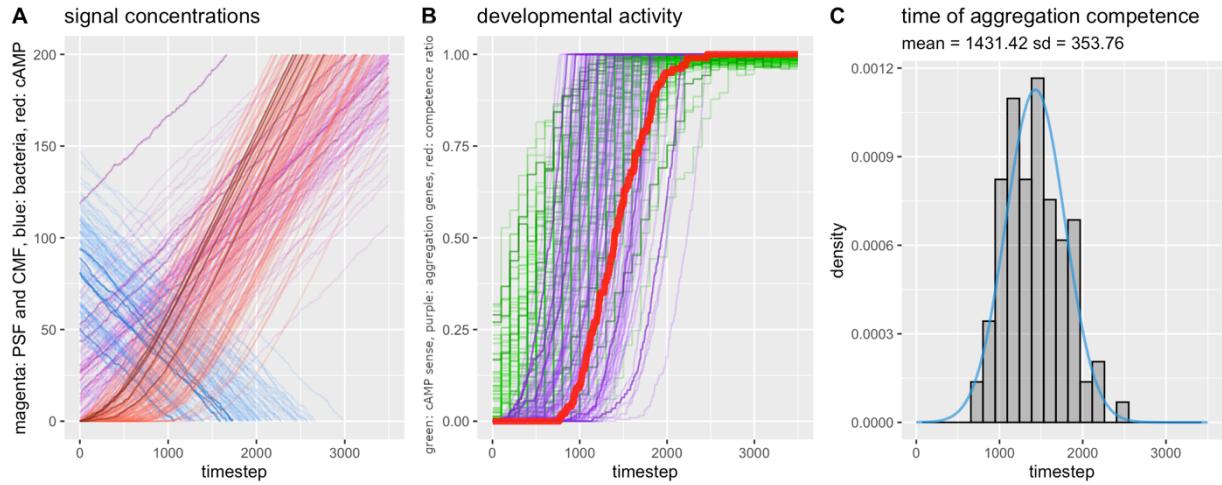


Figure 1C-12: The entry of cells into development is desynchronized by variations in environmental cues despite identical signal sensitivity. **A:** Concentration of PSF and Bacteria vary among cells. This causes certain cells beginning to produce cAMP at a later stage than others. **B:** Due to varying cAMP levels, the activity of genetic mechanisms involved in early development also becomes heterogeneous. **C:** cells gain aggregation competence at significantly varying times.

In the biologically most realistic scenario, cell-cell heterogeneities in signal responsiveness and environmental heterogeneity exist simultaneously. When cells make decisions independently, this causes very large differences in the timing of aggregation competence (**Fig 1C-13**).

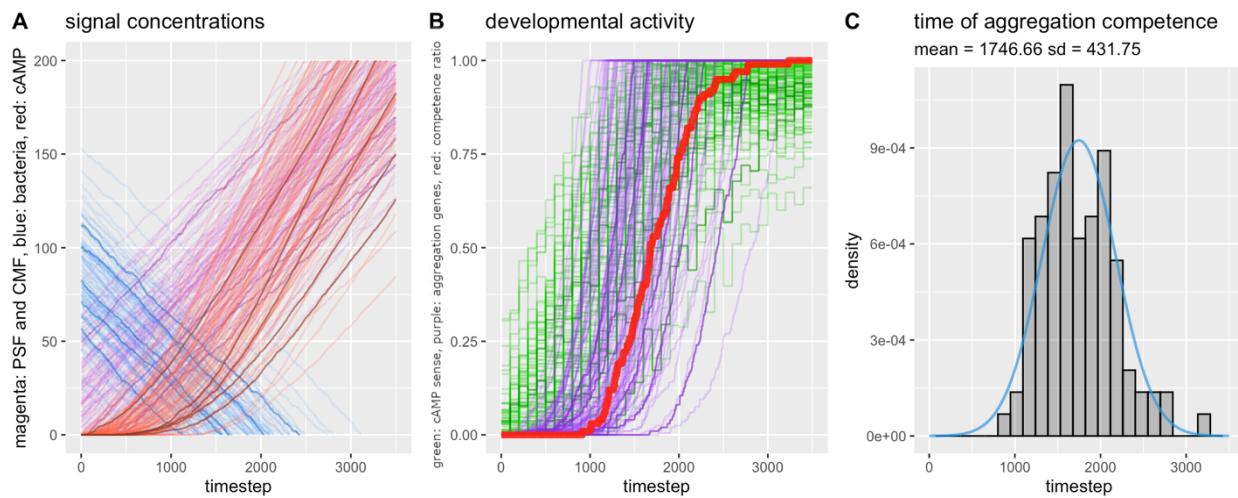


Figure 1C-13: The entry of cells into development is desynchronized by variations in signal sensitivity.

These results suggest that variation in the environment and signal responsiveness cause cells to begin development at significantly different times. During *Dictyostelium* development, the adequate response to signalling molecules will change with time. For example, during early development, cAMP should induce chemotaxis, while in later development, cAMP should induce prespore differentiation (Schaap 2016). As such, if cells begin development at different times, the resulting desynchronization of signal responses will undoubtably hamper morphogenesis and cell differentiation.

In the second stage of modelling, the goal was to investigate whether the export of cAMP will facilitate the synchronization of cells. As most of the cAMP produced by group 4 Dictyostelia is exported, the model was modified to diffuse cAMP.

Even though populations had highly variable signal sensitivity, CMF concentrations, and Bacteria concentrations, when produced cAMP is exported, cells with high cAMP production can stimulate cells exposed to low cAMP, reducing heterogeneity in signal levels, and synchronizing the entry into aggregation (**Fig. 1C-14**).

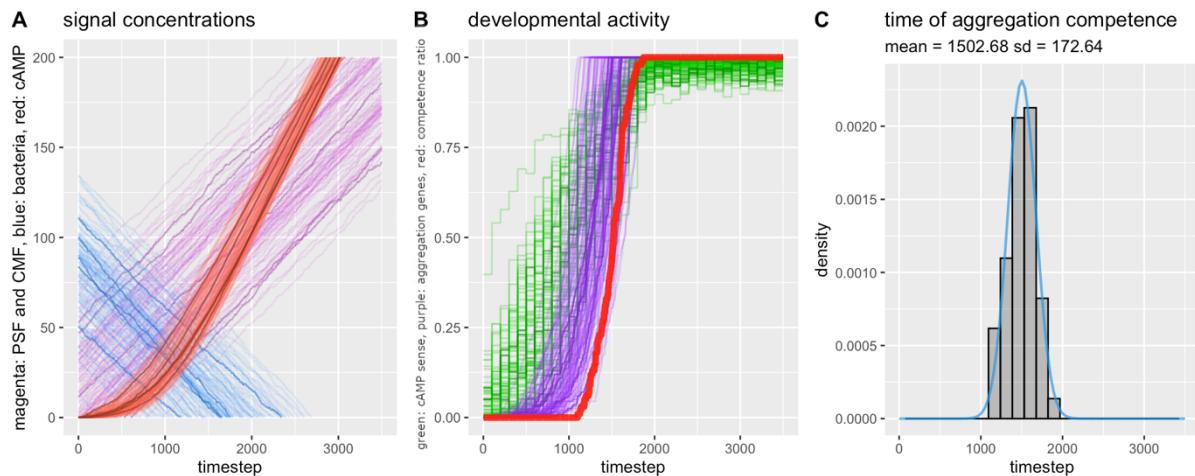


Figure 1C-14: The entry of cells into development is synchronized by exported cAMP despite environmental heterogeneities and cell-cell heterogeneities in signal sensitivity. A: Concentration of PSF and Bacteria vary among cells. However, the diffusion of cAMP can homogenize the exposure of individual cells to cAMP. B: Due to homogenous cAMP levels, the activity of genetic mechanisms involved in early development becomes synchronized. C: the time of aggregation competence is similar among cells. Note the lack of “late developers” which would be expected when certain cells have low signal sensitivity

This result suggests that when cell-cell signalling is coupled with a positive feedback loop, noise arising from environmental heterogeneity and variations signal sensitivity can be effectively buffered, allowing the population to make collective deterministic cell fate choices.

Buffering Noise: Conclusions

Dictyostelium developmental initiation involves buffering noise to enable the progression of deterministic developmental choices. During the unicellular stage of development, stochastic cell-cell differences arise from intrinsic noise in transcription, and extrinsic noise originating from heterogeneities in the microenvironment of cells. To overcome these differences and synchronize cells for later stages of development, amoeba utilize cell-cell signalling combined to positive feedback loops. What can observations in this system suggest about the buffering of noise in general?

- 1) *Cell-cell signalling is the key to overcoming heterogeneity, as it provides all cells within the population with the necessary information to make correct decisions.*

These mechanisms rely on the fact that information gained from multiple cells can generate more stable outcomes. The early embryonic development of *drosophila* typifies this technique. While the level of transcription in identical developmental patterning gene loci can vary by as much as 45% between nuclei, the early *drosophila* embryo can develop robustly because the embryonic tissue forms a cytoplasmic syncytium, meaning that mRNA from high and low transcribing nuclei are transported into a shared cytoplasm, resulting in a spatiotemporal averaging of mRNA concentrations (Little et al 2013).

- 2) *Positive feedback loops are an important motif of regulatory systems designed to drive decisive cell fate decisions in noisy environments*

The striking similarity between the use of positive feedback loops to drive synchronous entry into development in *Dictyostelia*, and the use of positive feedback loops to drive robust tumorigenicity and metastasis in basal-like breast cancer suggest that this motif of decision-making systems enables cells to coordinate actions when environments are highly heterogeneous.

Section 2: The use of noise in cell fate differentiation

Introduction

The differentiation of prestalk and prespore cells takes place in the mound. How can a homogenous population of cells establish robust ratios of diverging cell types?

The model of lineage priming followed by salt-and-pepper differentiation (LP+SP) provides a mechanism by which a population of cells exposed to identical signals can break symmetry. In this model, non-genetic heterogeneity is accumulated throughout the lifetime of cells, which results in varying sensitivity to a morphogenic signal. When cells are exposed to identical morphogen levels, only the more sensitive “lineage primed” cells react by differentiating. As lineage priming occurs due to stochastic effects, primed cells will be scattered throughout the population. Therefore, differentiated cells will appear at random locations, a spatial pattern known as “salt-and-pepper” differentiation (Chattwood & Thompson 2011, Chattwood et al 2013).

Section 2A:

Evidence of salt-and-pepper differentiation in *Dictyostelium*

A growing body of evidence suggests that *Dictyostelium* cell type proportioning occurs using LP+SP. As DIF-1 and cAMP (the morphogens promoting prestalk and prespore fates respectively) diffuse throughout the mound, all cells are exposed to homogenous signalling microenvironments (Chattwood & Thompson 2011, Schaap 2011, Araki & Saito 2019). Stevense et al (2010), visualized morphogen induced transcription, and found that as morphogen doses increased, more cells manifested a transcriptional response, but the response strength did not increase substantially. These observations favour a view where individual cells have different sensitivity thresholds for morphogens. Finally, by using Latrunculin-A to block the movement of cells in the mound, Thompson et al (2004) demonstrated that PSP and PST cells differentiate at scattered positions, which is the spatial pattern expected under an LP+SP system.

These results suggest that cell fate differentiation is generated using heterogeneities in signal responsiveness. From here the question arises: What are the sources of non-genetic heterogeneity in *Dictyostelium* lineage priming?

Section 2B: Sources of stochasticity

Although the expression of PST and PSP specific gene markers begins 11 hours after developmental initiation (Rosengarten et al 2015), cell fate biases have already been established at earlier stages of development. This section will detail some of the cell parameters that have been shown to affect cell fate.

Variations in cellular ATP levels are consequential throughout development. Using sensor probes to observe ATP levels in developing cells, Hiraoka et al (2019) find that cells with higher levels of ATP are biased towards the stalk fate, while cells with lower ATP levels are more likely to become spores. Additionally, manually reducing ATP levels by inhibiting production resulted in a significant spore bias.

Cell-cell variation in intracellular Ca^{2+} levels are observable immediately after starvation and have a significant impact on cell fate throughout development. Before starvation, all cells move in random directions, with a unimodal distribution of speed. Following the starvation, two classes of cells emerge. Cells with low calcium (future spores, ~81% of cells) move slower than cells with high calcium (future stalk, 19 % of cells) (Goury-Sistla et al 2012).

The phase of the cell cycle that a given cell was in at the time of starvation also influences the initial cell type choice. Cells that begin to starve in S and early G_2 become primarily stalk cells, while cells in late G_2 will most likely become spore cells (Jang & Gomer 2011) (**Fig. 2B-1**). The cell cycle phase has also been shown to significantly modulate the sensitivity to morphogens. For example, Thompson & Kay (2000) found that half maximal stalk cell induction required only 4nM DIF-1 for S and early G_2 cells, but 10nM for late G_2 cells. Cell cycle phase and Ca^{2+} concentrations are also linked, as Cells in S and early G_2 have high Ca^{2+} , while Cells in mid to late G_2 have low Ca^{2+} (Jang & Gomer 2011, Chattwood & Thompson 2011).

If cell cycle position is an important driver of cell fate decisions, robust and reproducible proportioning can only occur if cell cycle phases are predictably distributed within the population when differentiation is induced. Is the cell cycle a reliable source of stochasticity within a population?

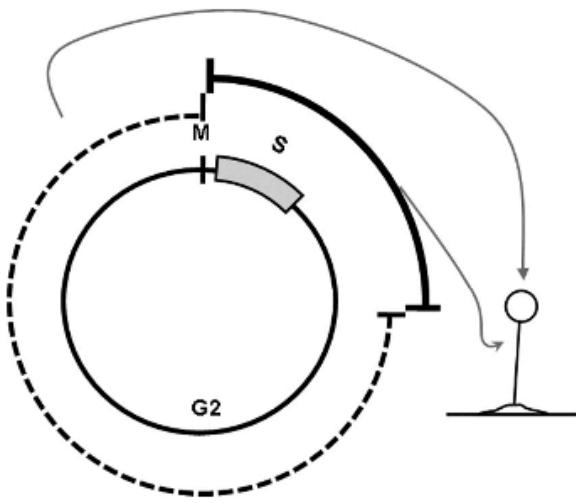


Figure 2B-1: The cell cycle-dependent model of initial cell type choice in *Dictyostelium*. Cells in early development make an initial cell type choice that depends on the phase of the cell cycle at the time of starvation. Cells in S and early G₂ phases are likely to become prestalk cells (solid line). Cells in mid to late G₂ are likely to become prespore cells. Source: Jang & Gomer 2011

Section 2C: Recent work by the Thompson lab and collaborators

Cell cycle heterogeneity is an important source of robustness

To answer this question, Gruenheit et al (2018) investigated whether the *Dictyostelium* cell cycle lengths exhibit sufficient levels of cell-cell heterogeneity to desynchronize the population.

When cell cycle length was measured using time lapse microscopy, they found very little correlation between mother and daughter cell cycle lengths. Using the microscopy data, a mathematical model was built that was used to determine the amount of time required to completely desynchronize a population that clonally arises from a single cell. The model revealed that by the time 5 divisions had occurred (roughly 40 hours), the population had become desynchronized. As typical fruiting body formation involves 10^4 - 10^6 cells, cell cycle positions would be completely randomized by the time development initiates.

The randomization of cell cycle positions relative to each other would imply that the cell fate bias of any single observed cell is stochastic, but the population has ample amounts of cells in each phase, leading to predictable and relatively stable proportions at the population level (**Fig. 2C-1**).

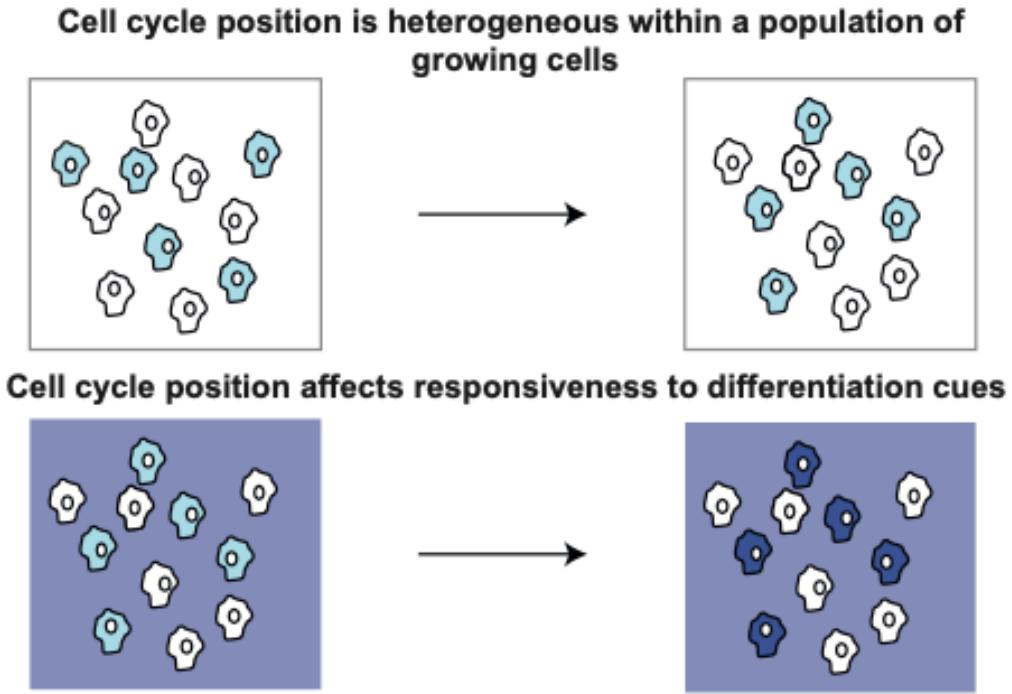


Figure 2C-1: cell cycle heterogeneity can vary the response threshold to morphogens. If the distribution of cells within cell cycle phases is predictable at the population level, this can enable the robust and consistent generation of cell type ratios. Source: Gruenheit et al 2018

The use of desynchronized cytokinesis to generate functional heterogeneity is not unique to *Dictyostelia*. For example, Roeder et al (2010) utilized a combination of microscopy and computer modelling to investigate how cell size patterning in the sepal epidermis of *Arabidopsis thaliana* can be explained by the variability in the times when cells divide, and the time at which they stop dividing, concluding that stochastic decisions of individual cells enables the robust production of cell size patterns at the population level during development.

Stochastic cell fate differentiation integrates multiple independent sources of noise for increased robustness

While the above findings highlight the ability of cell cycle desynchronization to produce reliable outcomes at the population level, it does not explain how *Dictyostelium* populations successfully break symmetry when cell cycle progression is blocked, or cell cycle states have been fully synchronized. For example, isolated cells in G₂ will still form more than 7% stalk cells (Gruenheit et al 2018). This suggests that a cell-cycle independent source of variation is integrated into cell fate decisions.

To pinpoint the independent source of variation Salvidge et al (manuscript in preparation) started by identifying genes with a large degree of cell-cell variation. Within these variant genes, AUROC analyses were utilized to identify genes that were associated with specific phases of the cell cycle, leaving a limited subset of genes with cell-cycle independent stochasticity (CCIS). To identify whether the newly identified CCIS is important for symmetry breaking, the authors set out to create a *Dictyostelium* strain where CCIS could be disrupted.

Because specific histone modifications have been associated with changes in burst frequency, and thus increased heterogeneity, the H3K4 mono, di, or tri-methylation (H3K4me1-3) system which is dependent on Set1/COMPASS was selected, and strains where generated Set1 was disrupted by homologous recombination.

Using scRNA data, the scientists found that Set1 disruption caused many of the cell-cycle dependent, and cell-cycle independent genes to change expression levels. Additionally, GFP reporters showed that increases in expression levels were due to increased bursting frequency. As most differentially expressed genes increased their expression levels, and increased bursting frequency is associated with lower levels of heterogeneity, this suggested that the disruption of Set1 reduces population level variation in gene expression. Crucially, the authors also found that Set1 defective strains had altered cell fate decisions. While Set⁺ populations could develop with minor abnormalities, when developed in a chimera with wild-type strains, Set⁻ strains tended to occupy the posterior and collar regions of the slug, and the upper and lower cups within the fruiting body, suggesting a stalky bias. Transcriptomic profiling of developing Set1⁻ mutants also showed that prespore genes were underexpressed, while prestalk genes were overexpressed compared to wild type.

To identify why Set1 defects impair cell type proportioning, the sensitivity to the prestalk inducing DIF-1 was investigated. Using GFP reporters of DIF-1 sensitive prestalk associated ecmA and ecmB genes, it was found that Set1⁻ cells had significantly higher DIF-1 sensitivity, and that the proportion of cells that react to DIF-1 is much higher when Set1 is disrupted, confirming that the cell-cycle independent stochasticity seen in wild-type strains is partially caused by Set1.

Finally, the researchers disrupted both sources of stochasticity. When Set1^- cells were grown at low temperatures, blocking cell cycle progression, development completely stalled, with only a fraction of samples generating fruiting bodies.

These results highlight the fact that having sufficient sources of variability is crucial for the successful completion of developmental programs that generate diverging cell fates from seemingly homogenous cell populations

Section 2D: Correcting mistakes with lateral inhibition

What is lateral inhibition?

Lateral inhibition involves a population of initially equivalent cells competing for fates during differentiation, during which each cell type produces substances that inhibit the differentiation of their neighbours into the same type (Barad et al 2010). This mechanism is used to correct the ratio of PST and PSP cells in *Dictyostelium* when initial proportions are not optimal, or when slugs are amputated in such a way that removes one of the cell types (Rafols et al 2001, Mohri et al 2020).

How is lateral inhibition implemented?

In this system, DIF is manufactured by PSP cells and degraded by PST cells. DIF signalling induces the expression of genes specific to the PST fate, while also inhibiting cAMP sensitivity, blocking the PSP fate. cAMP is made by PST cells and degraded by PSP cells. The activity of cAMP signalling increases the expression of PSP specific genes, while also inhibiting sensitivity to DIF, effectively blocking the PST fate. (Kay & Thompson 2001, Kay & Thompson 2009, Pineda et al 2015) (**Fig. 2D-1**).

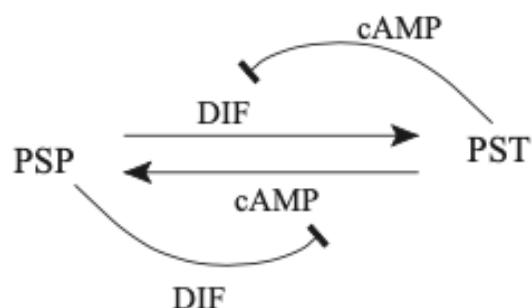


Figure 2D-1: Lateral inhibition of cell types via DIF and cAMP. Source: Pineda et al 2015

What are the effects of lateral inhibition?

Lateral inhibition allows the population to correct perturbations to the PST:PSP ratio. For example, if PST:PSP ratio is disrupted by lowering the number of PSP cells, DIF levels decrease due to a lack of production and cAMP levels increase due to decreased breakdown. Therefore, some PST cells that were previously exposed to high DIF and low cAMP levels are now exposed to lower DIF and higher cAMP levels, allowing them to switch to the PSP fate. As the transdifferentiation of PST to PSP cells is occurring, DIF levels begin to rise, and cAMP levels begin to fall, returning the system to a balanced state.

Additionally, because DIF and cAMP diffuse to all parts of the organism, lateral inhibition can accurately regulate cell type ratios, irrespective of the number of cells. This is especially important in *Dictyostelium*, where colony sizes can vary by four magnitudes (Schilde et al 2014)

Lateral inhibition in other organisms

A similar type of lateral inhibition mechanism is used to establish cell fates in animals. During gonadogenesis in *C. elegans*, homogenous cells differentiate into two distinct types, anchor cells (AC) and ventral uterine cells (VU) utilizing the LIN-12 signalling pathway. Cells carry LIN-12 receptors on their surface, and produce its ligand LAG-2. Exposure to LAG-2 stimulates the LIN-12 pathway, which leads to LIN-12 receptors being produced, and decreases the production of LAG-2. If cell A has slightly more LIN-12 receptors than its neighbour B, this difference will be amplified because cell A will become more and more sensitive to LIN-12, while producing less and less LAG-2, lowering the LAG-2 exposure of cell B. As higher levels of LIN-12 lead to the VU fate, while low levels lead to the AC fate, this system will reliably produce a heterogenous mixture of AC and VU cells (Greenwald 1998) (**Fig. 2D-2**).

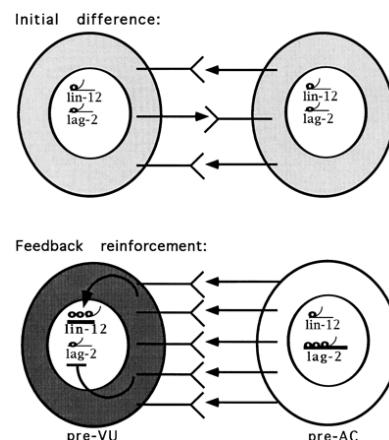


Figure 2D-2: Lateral inhibition allows the generation of diverging cell fates in *C. elegans* gonadogenesis. Initial differences in LIN-12 sensitivity are amplified using a feedback loop to produce large differences in LIN-12 pathway activity levels between cells, which then lead to diverging VU and AC fates being adopted. Source: Greenwald 1998

Noise in cell fate differentiation: Conclusions

Dictyostelium cell fate differentiation represents a crucial example of the use of noise during stochastic developmental decisions. During aggregation, cells utilize the slight differences in physiological states and cell cycle phase to potentiate their response to morphogens, priming cells to adapt diverging lineages during the mound phase. After the initial proportion of cells were established, lateral inhibition is utilized to correct for any imbalances that may have appeared. What can the process of differentiation in *Dictyostelium* suggest about the use of noise to generate divergent cell fates from homogenous populations?

1) *Stochasticity can be an essential starting point for developmental stages*

The facts that *Dictyostelium* integrates information from multiple independent sources of noise, and development fails when both sources of stochasticity are abolished indicates that noise can be a fundamental requirement for the completion of developmental programs.

2) *When decisions are based on stochastic inputs, systems must be able to detect and compensate possible biases affecting the stochastic system.*

All biologically stochastic systems are subject to environmental effects that can bias the state of the population. If no mechanisms exist to detect and correct these biases, later developmental stages can be expected to fail.

Future research

There are many stochastic mechanisms involved in development that have not been uncovered. As genetic modification techniques gain increased precision and transcriptomic methods gain increased temporal resolution, *Dictyostelium discoideum* will continue to provide significant insight into the roles of noise in development. Within *Dictyostelium* research, Future experiments will need to uncover whether the rare successful development of cell cycle arrested Set1⁻ strains is due to inadequacies in the experimental tools used to abolish cell-cycle and Set1 based variation, or due to currently unknown sources of noise utilized during the developmental process. The application of stochasticity perspectives to cancer is also a highly promising field. For example, future experiments could investigate how the use of drugs to inhibit components of positive feedback loops in cancers affects patient outcome.

References

- Altschuler, S. and Wu, L., 2010. Cellular Heterogeneity: Do Differences Make a Difference? *Cell*, 141(4), pp.559-563.
- Araki, T. and Saito, T., 2019. Small molecules and cell differentiation in Dictyostelium discoideum. *The International Journal of Developmental Biology*, 63(8-9-10), pp.429-438.
- Balázsi, G., van Oudenaarden, A. and Collins, J., 2011. Cellular Decision Making and Biological Noise: From Microbes to Mammals. *Cell*, 144(6), pp.910-925.
- Barad, O., Rosin, D., Hornstein, E. and Barkai, N., 2010. Error Minimization in Lateral Inhibition Circuits. *Science Signaling*, 3(129), pp.ra51-ra51.
- Bozzaro, S., 2019. The past, present and future of Dictyostelium as a model system. *The International Journal of Developmental Biology*, 63(8-9-10), pp.321-331.
- Chattwood, A. and Thompson, C., 2011. Non-genetic heterogeneity and cell fate choice in Dictyostelium discoideum. *Development, Growth & Differentiation*, 53(4), pp.558-566.
- Chattwood, A., Nagayama, K., Bolourani, P., Harkin, L., Kamjoo, M., Weeks, G. and Thompson, C., 2013. Developmental lineage priming in Dictyostelium by heterogeneous Ras activation. *eLife*, 2.
- Consalvo, K., Rijal, R., Tang, Y., Kirolos, S., Smith, M. and Gomer, R., 2019. Extracellular signaling in Dictyostelium. *The International Journal of Developmental Biology*, 63(8-9-10), pp.395-405.
- Dong, P. and Liu, Z., 2017. Shaping development by stochasticity and dynamics in gene regulation. *Open Biol.* 7: 170030.
- Eldar, A. and Elowitz, M., 2010. Functional roles for noise in genetic circuits. *Nature*, 467(7312), pp.167-173.
- Eling, N., Morgan, M. and Marioni, J., 2019. Challenges in measuring and understanding biological noise. *Nature Reviews Genetics*, 20(9), pp.536-548.
- Elowitz, M., Levine, A., Siggia, E. and Swain, P., 2002. Stochastic Gene Expression in a Single Cell. *Science*, 297(5584), pp.1183-1186.
- Fujimori, T., Nakajima, A., Shimada, N. and Sawai, S., 2019. Tissue self-organization based on collective cell migration by contact activation of locomotion and chemotaxis. *Proceedings of the National Academy of Sciences*, 116(10), pp.4291-4296.
- Giacomantonio, C. and Goodhill, G., 2010. A Boolean Model of the Gene Regulatory Network Underlying Mammalian Cortical Area Development. *PLoS Computational Biology*, 6(9), p.e1000936.
- Goury-Sistla, P., Nanjundiah, V. and Pande, G., 2012. Bimodal distribution of motility and cell fate in Dictyostelium discoideum. *The International Journal of Developmental Biology*, 56(4), pp.263-272.

- Greenwald, I., 1998. LIN-12/Notch signaling: lessons from worms and flies. *Genes & Development*, 12(12), pp.1751-1762.
- Gruenheit, N., Parkinson, K., Brimson, C., Kuwana, S., Johnson, E., Nagayama, K., Llewellyn, J., Salvidge, W., Stewart, B., Keller, T., van Zon, W., Cotter, S. and Thompson, C., 2018. Cell Cycle Heterogeneity Can Generate Robust Cell Type Proportioning. *Developmental Cell*, 47(4), pp.494-508.e4.
- Hiraoka, H., Nakano, T., Kuwana, S., Fukuzawa, M., Hirano, Y., Ueda, M., Haraguchi, T. and Hiraoka, Y., 2020. Intracellular ATP levels influence cell fates in *Dictyostelium discoideum* differentiation. *Genes to Cells*, 25(5), pp.312-326.
- Jain, R. and Gomer, R., 1994. A developmentally regulated cell surface receptor for a density-sensing factor in *Dictyostelium*. *Journal of Biological Chemistry*, 269(12), pp.9128-9136.
- Jang, W. and Gomer, R., 2011. Initial Cell Type Choice in *Dictyostelium*. *Eukaryotic Cell*, 10(2), pp.150-155.
- Kawabe, Y., Du, Q., Schilde, C. and Schaap, P., 2019. Evolution of multicellularity in *Dictyostelia*. *The International Journal of Developmental Biology*, 63(8-9-10), pp.359-369.
- Kay, R. and Thompson, C., 2001. Cross-induction of cell types in *Dictyostelium*: evidence that DIF-1 is made by prespore cells. *Development*, 128(24), pp.4959-4966.
- Kay, R. and Thompson, C., 2009. Forming Patterns in Development without Morphogen Gradients: Scattered Differentiation and Sorting Out. *Cold Spring Harbor Perspectives in Biology*, 1(6), pp.a001503-a001503.
- Little, S., Tikhonov, M. and Gregor, T., 2013. Precise Developmental Gene Expression Arises from Globally Stochastic Transcriptional Activity. *Cell*, 154(4), pp.789-800.
- Loomis, W., 2014. Cell signaling during development of *Dictyostelium*. *Developmental Biology* 391 (2014) 1–16
- Loomis, W., 2015. Genetic control of morphogenesis in *Dictyostelium*. *Developmental Biology* 402 (2015) 146–161
- Losick, R. and Desplan, C., 2008. Stochasticity and Cell Fate. *Science*, 320(5872), pp.65-68.
- McAdams, H. and Shapiro, L., 1995. Circuit simulation of genetic networks. *Science*, 269(5224), pp.650-656.
- Mohri, K., Tanaka, R. and Nagano, S., 2020. Live cell imaging of cell movement and transdifferentiation during regeneration of an amputated multicellular body of the social amoeba *Dictyostelium discoideum*. *Developmental Biology*, 457(1), pp.140-149.

- Muramoto, T., Iriki, H., Watanabe, J. and Kawata, T., 2019. Recent Advances in CRISPR/Cas9-Mediated Genome Editing in Dictyostelium. *Cells*, 8(1), p.46.
- Nicol, A., Rappel, W., Levine, H. and Loomis, W., 1999. Cell-sorting in aggregates of Dictyostelium discoideum. *Journal of Cell Science*, 112(22), pp.3923-3929.
- Pineda, M., Weijer, C. and Eftimie, R., 2015. Modelling cell movement, cell differentiation, cell sorting and proportion regulation in Dictyostelium discoideum aggregations. *Journal of Theoretical Biology* 370 (2015) 135–150
- Radisavljevic, Z., 2012. AKT as locus of cancer positive feedback loops and extreme robustness. *Journal of Cellular Physiology*, 228(3), pp.522-524.
- Ràfols, I., Sawada, Y., Amagai, A., Maeda, Y. and MacWilliams, H., 2001. Cell type proportioning in Dictyostelium slugs: lack of regulation within a 2.5-fold tolerance range. *Differentiation*, 67(4-5), pp.107-116.
- Roeder, A., Chickarmane, V., Cunha, A., Obara, B., Manjunath, B. and Meyerowitz, E., 2010. Variability in the Control of Cell Division Underlies Sepal Epidermal Patterning in *Arabidopsis thaliana*. *PLoS Biology*, 8(5), p.e1000367.
- Rosengarten, R., Santhanam, B., Fuller, D., Katoh-Kurasawa, M., Loomis, W., Zupan, B. and Shaulsky, G., 2015. Leaps and lulls in the developmental transcriptome of Dictyostelium discoideum. *BMC Genomics*, 16(1).
- Salvidge, W., Brimson, C., Gruenheit, N., Huang, L., Pears, C., Wolf, J. and Thompson, C., n.d. Stochastic gene expression facilitates robust symmetry breaking, cell fate choice and cell type proportioning (working title).
- Schaap, P. and Schilde, C., 2018. Encystation: the most prevalent and underinvestigated differentiation pathway of eukaryotes. *Microbiology*, 164(5), pp.727-739.
- Schaap, P., 2007. Evolution of size and pattern in the social amoebas. *BioEssays*, 29(7), pp.635-644.
- Schaap, P., 2011. Evolutionary crossroads in developmental biology: Dictyostelium discoideum. *Development*, 138(3), pp.387-396.
- Schaap, P., 2016. Evolution of developmental signalling in Dictyostelid social amoebas. *Current Opinion in Genetics & Development* 2016, 39:29–34
- Schilde, C., Skiba, A. and Schaap, P., 2014. Evolutionary reconstruction of pattern formation in 98 Dictyostelium species reveals that cell-type specialization by lateral inhibition is a derived trait. *EvoDevo*, 5(1), p.34.
- Shmulevich, I., Dougherty, E. and Zhang, W., 2002. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11), pp.1778-1792.
- Siegert, F. and Weijer, C., 1995. Spiral and concentric waves organize multicellular Dictyostelium mounds. *Current Biology*, 5(8), pp.937-943.

- Singh, A. and Soltani, M., 2013. Quantifying Intrinsic and Extrinsic Variability in Stochastic Gene Expression Models. *PLoS ONE*, 8(12), p.e84301.
- Stevense, M., Muramoto, T., Müller, I. and Chubb, J., 2010. Digital nature of the immediate-early transcriptional response. *Development*, 137(4), pp.579-584.
- Swanson, A., Vadell, E. and Cavender, J., 1999. Global distribution of forest soil dictyostelids. *Journal of Biogeography*, 26(1), pp.133-148.
- Thompson, C. and Kay, R., 2000. Cell-Fate Choice in Dictyostelium: Intrinsic Biases Modulate Sensitivity to DIF Signaling. *Developmental Biology*, 227(1), pp.56-64.
- Thompson, C., Reichelt, S. and Kay, R., 2004. A demonstration of pattern formation without positional information in Dictyostelium. *Development, Growth and Differentiation*, 46(4), pp.363-369.
- Trenchard, H., 2019. Cell pelotons: A model of early evolutionary cell sorting, with application to slime mold *Dictyostelium discoideum*. *Journal of Theoretical Biology*, 469, pp.75-95.
- Urban, E. and Johnston, R., 2018. Buffering and Amplifying Transcriptional Noise During Cell Fate Specification. *Frontiers in Genetics*, 9.
- Viney, M. and Reece, S., 2013. Adaptive noise. *Proc R Soc B* 280: 20131104.
- Wu, X., Li, X., Fu, Q., Cao, Q., Chen, X., Wang, M., Yu, J., Long, J., Yao, J., Liu, H., Wang, D., Liao, R. and Dong, C., 2017. AKR1B1 promotes basal-like breast cancer progression by a positive feedback loop that activates the EMT program. *Journal of Experimental Medicine*, 214(4), pp.1065-1079.

Appendix: R code.

```

library(pheatmap)
library(ggplot2)
library(reshape2)
library(ggplotify)
library(ggpubr)

##### SMALL FUNCTIONS #####
# this function adds a line to an existing ggplot object
add_line_to_plot <- function(existing_plot, gdf, first, second, col, a){
  existing_plot + geom_line(data = gdf, aes_string(x=first, y=second), color = col, alpha = a)
}

# this function wraps 1d coordinates, such that if 10 cells exist, the neighbors of cell 10 will be 9 and 1
c_wrap <- function(input_vec, max_val){
  l = length(input_vec)
  wrapped <- numeric(length = l)
  for (i in 1:l){
    if (input_vec[i] == 0){
      wrapped[i] <- max_val
    } else if ( input_vec[i]>max_val || input_vec[i]<1 ){
      wrapped[i] <- input_vec[i]%%max_val
    } else {
      wrapped[i] <- input_vec[i]
    }
  }
  return(wrapped)
}

# function that implements a random failure with chance x
prob_fail <- function(input_state, chance){
  if (runif(1, 0, 100) < chance){
    return(input_state*0)
  } else {
    return(input_state*1)
  }
}

# function that acts as an and gate
and_gate <- function(input_vec){
  g_state <- 0
  if (input_vec[1] == 1 && input_vec[2] == 1){
    g_state <- 1
  }
  return(g_state)
}

# function to calculate the input processing part of the circuit
cir_input <- function(input_vec, response_chance){
  psf_state <- input_vec[1]
  bac_state <- input_vec[2]
  cmf_state <- input_vec[3]

  psf_out <- prob_fail(psf_state, response_chance)
  bac_out <- prob_fail(!bac_state, response_chance)
  cmf_out <- prob_fail(cmf_state, response_chance)

  return(c(psf_out, bac_out, cmf_out))
}

# function to calculate the and gates reacting to the input
cir_andgates <- function(input_vec){
  # and gate 1 combining the signal from PSF and BAC
  g1_state <- and_gate(input_vec[c(1, 2)])
  # and gate 2 combining the signal from CMF and BAC
  g2_state <- and_gate(input_vec[c(2, 3)])

  return(c(g1_state, g2_state))
}

# function to calculate the latch behaviour of g3
cir_latch <- function(input_vec){
  output_state <- 0
  if (sum(input_vec) > 0){
    output_state <- 1
  }
  return(output_state)
}

# function to calculate the activity of PKA, ACA, and CAR1
cir_PKA ACA CAR <- function(input_vec){
  PKA_state <- input_vec[1]
  gateH_state <- input_vec[2]
  cAMP_state <- input_vec[3]

  ACA_state <- ceiling(sum(PKA_state, gateH_state)/2)
  CAR1_state <- ceiling(sum(PKA_state, cAMP_state)/2)
  return(c(ACA_state, CAR1_state))
}

# function to calculate the state of gate H, which completes the feedback loop
cir_gateH <- function(input_vec){

```

```

g3_state <- input_vec[1]
CAR1_state <- input_vec[2]
return(and_gate(c(g3_state, CAR1_state)))
}

# function integrating the whole circuit
# c_signal is the chance that the cell will respond to a signal, c_internal is the chance that a component
fails
circuit_sim <- function(input_cell_vec, c_signal, c_internal){
  ## one clock cycle
  out_state <- input_cell_vec
  # input processing

  # gate 1 and gate 2 processing
  gate_outputs <- cir_andgates(cir_input(out_state[1:3], c_signal))
  out_state[4] <- gate_outputs[1]; out_state[5] <- gate_outputs[2];
  # probabilistic fail
  out_state[4] <- prob_fail(out_state[4], c_internal)
  out_state[5] <- prob_fail(out_state[5], c_internal)

  # gate 3 processing
  out_state[6] <- cir_latch(c(out_state[5], out_state[6]))

  # PKA ACR CAR processing
  out_state[7] <- out_state[4]
  PKA_ACA_CAR_outputs <- cir_PKA_ACA_CAR(c(out_state[4], out_state[11], prob_fail(out_state[12],
c_signal)))
  out_state[8] <- PKA_ACA_CAR_outputs[1]; out_state[9] <- PKA_ACA_CAR_outputs[2]
  # probabilistic fail
  out_state[7] <- prob_fail(out_state[7], c_internal)
  out_state[8] <- prob_fail(out_state[8], c_internal)
  out_state[9] <- prob_fail(out_state[9], c_internal)

  # cAMP
  out_state[10] <- out_state[8]
  # probabilistic fail
  out_state[10] <- prob_fail(out_state[10], c_internal)

  # gate H state
  out_state[11] <- cir_gateH(c(out_state[6], out_state[9]))

  return(out_state)
}

# this function calculates the inputs to the circuit, based on the concentration of the signalling molecules
signal_sensors <- function(input_cell_vec){
  # return signal outputs
  #PSF output
  if (runif(1, 0, 100) < input_cell_vec[13]){
    input_cell_vec[1] <- 1
  } else {
    input_cell_vec[1] <- 0
  }
  #BAC output
  if (runif(1, 0, 100) < input_cell_vec[14]){
    input_cell_vec[2] <- 1
  } else {
    input_cell_vec[2] <- 0
  }
  #CMF output
  if (runif(1, 0, 100) < input_cell_vec[15]){
    input_cell_vec[3] <- 1
  } else {
    input_cell_vec[3] <- 0
  }
  #cAMP output
  if (runif(1, 0, 100) < input_cell_vec[16]){
    input_cell_vec[12] <- 1
  } else {
    input_cell_vec[12] <- 0
  }
  return(input_cell_vec)
}

# this function implements the changes the level of signalling molecules due to growth, bacteria, and cAMP
production
signal_levels <- function(input_cell_vec, PSF_CMF_up, BAC_down, cAMP_up, sd_ratio){
  # implement PSF increase
  input_cell_vec[13] <- input_cell_vec[13] + rnorm(1, mean = PSF_CMF_up, sd = PSF_CMF_up*sd_ratio)
  # implements BAC decrease
  input_cell_vec[14] <- input_cell_vec[14] - rnorm(1, mean = BAC_down, sd = BAC_down*sd_ratio)
  if (input_cell_vec[14] < 0 ){
    input_cell_vec[14] <- 0
  }
  # implements CMF increase when BAC is low
  if (input_cell_vec[14] < 5){
    input_cell_vec[15] <- input_cell_vec[15] + rnorm(1, mean = PSF_CMF_up*0.8, sd = PSF_CMF_up*sd_ratio)
  }

  # implements cAMP increase if cAMP is active
  if (input_cell_vec[10] == 1){
    # basal increase
    input_cell_vec[16] <- (input_cell_vec[16] + PSF_CMF_up)
    # exponential increaese
    if(input_cell_vec[16] > 50) {
      input_cell_vec[16] <- input_cell_vec[16]*cAMP_up
    }
  }
}

```

```

    }
}

return(input_cell_vec)
}

# this function implements the diffusion of signalling molecules
signal_diffuse <- function(input_signalling_levels, diffusion_speed){
  # set up temporary vector to hold results
  start_vector <- input_signalling_levels
  # number of locations between which the molecules diffuse
  bins <- length(input_signalling_levels)
  # this vector keeps track of the
  diffused_vector <- vector( length = bins)
  for (i in 1:bins){
    # the molecules diffusing from bin i will land in the neighboring bins
    target_bins <- c_wrap(c(i-1, i+1), bins)
    # the molecules will unevenly split between the target bins
    diffused_vector[target_bins] <- diffused_vector[target_bins] + (start_vector[i]*diffusion_speed/2)
  }
  # subtract the diffusing molecules from the starting density
  start_vector <- start_vector*(1-diffusion_speed)
  # add in the molecules that diffused
  start_vector <- start_vector+ diffused_vector
  return(start_vector)
}

##### TIME DOMAIN BEHAVIOUR OF A SINGLE CELL #####
timesteps <- 500
row_names <- c("PSF", "BAC", "CMF", "G1", "G2", "G3",
             "PKA", "ACA", "CAR1", "cAMP_prod", "G4", "cAMP_sense",
             "PSF_conc", "BAC_conc", "CMF_conc", "cAMP_conc")

# set up cell state holder
cell_state <- matrix(0, nrow = 16, ncol = 1)
rownames(cell_state) <- row_names
# PSF CMF and cAMP start low and accumulate
cell_state[13] <- 5
cell_state[15] <- 0
cell_state[16] <- 0
# BAC starts at 100 and decreases
cell_state[14] <- 55

# set up matrix to hold cell states over time
cell_state_matrix <- matrix(0, nrow = 16, ncol = timesteps)
rownames(cell_state_matrix) <- row_names
colnames(cell_state_matrix) <- seq(1, timesteps)

for (i in 1:timesteps){
  # generate inputs to the cell circuit
  cell_state <- signal_sensors(cell_state)
  # run circuit simulation
  cell_state <- circuit_sim(cell_state, 5, 5)
  # increment levels of signalling molecules
  cell_state <- signal_levels(cell_state, 0.3, 0.6, 1.01, 1)
  # print results to see
  print(as.numeric((cell_state)))
  cell_state_matrix[,i] <- cell_state
}

activity <- as.ggplot(pheatmap(cell_state_matrix[1:12,], cluster_rows = FALSE, cluster_cols = FALSE,
show_colnames = F))
activity <- activity +labs(title = "          circuit component activation", x = "timestep")

cell_state_df <- as.data.frame(cell_state_matrix[13:16,])
cell_state_df <- as.data.frame(t(rbind(seq(1, timesteps), cell_state_df)))
signals <- ggplot(cell_state_df, aes(cell_state_df$`1`, cell_state_df$BAC_conc)) +
  geom_line(color = "dodgerblue3", size=2,) +
  geom_line(aes(cell_state_df$`1`, cell_state_df$cAMP_conc), color = "tomato3", size=2) +
  geom_line(aes(cell_state_df$`1`, cell_state_df$PSF_conc), color = "magenta3", size=2) +
  geom_line(aes(cell_state_df$`1`, cell_state_df$CMF_conc), color = "magenta2", size=2) +
  labs(title = "signal concentrations", x = "timestep", y = "magenta: PSF and CMF, blue: bacteria, red: cAMP")

ggarrange(signals, activity,
          labels = c("A", "B"),
          ncol = 2, nrow = 1)

##### BEHAVIOUR OF MULTIPLE CELLS #####
# this function sets up the initial cell state matrix
multisim_prep <- function(n_cells, quorum_starting, bac_starting, envir_var, signal_fail, signal_fail_var,
cell_fail, cell_fail_var){
  # set up cell state holder
  multi_cell_state <- matrix(0, nrow = 18, ncol = n_cells)
  rownames(multi_cell_state) <- c("PSF", "BAC", "CMF", "G1", "G2", "G3",
                                   "PKA", "ACA", "CAR1", "cAMP_prod", "gateH", "cAMP_sense",
                                   "PSF_conc", "BAC_conc", "CMF_conc", "cAMP_conc", "signal_fail",
"cell_fail")
  ## initial setup of signalling molecule levels
  # set starting values for PSF
  multi_cell_state[13,] <- (rnorm(n_cells, mean = quorum_starting, sd = envir_var))
  # set starting level for CMF, 0
  multi_cell_state[15,] <- 0
  # set starting levels for BAC
  multi_cell_state[14,] <- (rnorm(n_cells, mean = bac_starting, sd = envir_var))
  # set the chance that cells will not respond to signal
}

```

```

multi_cell_state[17,] <- abs(rnorm(n_cells, mean = signal_fail, sd = signal_fail_var))
multi_cell_state[18,] <- abs(rnorm(n_cells, mean = cell_fail, sd = cell_fail_var))
return(multi_cell_state)
}

# this function takes a starting cell state matrix and simulates the developing cells
multisim_simulate <- function(starting_multicell_state, timesteps, PSF_CMF_up, BAC_down, cAMP_prod,
signal_var, diffusion){
  # initialize starting state
  multi_cell_state <- starting_multicell_state
  # get the chance of reacting to signals and internal fails
  c_signal <- starting_multicell_state[17,]
  c_internal <- starting_multicell_state[18,]
  # get number of cells
  n_cells <- ncol(starting_multicell_state)
  # 3D matrix contains time series data for multi_cell_state
  multi_cell_state_time <- array(0, c(18, n_cells, timesteps))

  ## run simulation of multiple cells
  for (i in 1:timesteps){
    for (j in 1:n_cells){
      # separate out the state of an individual cell
      cell_state <- multi_cell_state[,j]
      # generate inputs to the cell circuit
      cell_state <- signal_sensors(cell_state)
      # run circuit simulation
      cell_state <- circuit_sim(cell_state, c_signal[j], c_internal[j])
      # increment levels of signalling molecules
      cell_state <- signal_levels(cell_state, PSF_CMF_up, BAC_down, cAMP_prod, signal_var)
      # write results back to the complete matrix
      multi_cell_state[,j] <- cell_state
    }
    # diffuse the signalling molecules
    # diffuse PSF
    multi_cell_state[13,] <- signal_diffuse(multi_cell_state[13,], diffusion)
    # diffuse CMF
    multi_cell_state[16,] <- signal_diffuse(multi_cell_state[16,], diffusion)
    # write the data to the time series array
    multi_cell_state_time[, , i] <- multi_cell_state
  }

  return(multi_cell_state_time)
}

##### ANALYZE THE LEVEL OF FEEDBACK LOOP ACTIVITY #####
multisim_analysis <- function(input_multi_cell_state_time, time_smooth){
  print("calculating statistics")
  # set the number of cycles the cAMP pathway should be active to initiate development
  dev_needed <- 200
  multi_cell_state_time <- input_multi_cell_state_time
  timesteps <- dim(multi_cell_state_time)[3]
  n_cells <- dim(multi_cell_state_time)[2]

  # create matrix to hold the coordinates of time slices
  time_slices <- matrix(ncol = 5, nrow = timesteps/5)
  for(i in 1:5){
    time_slices[,i] <- seq(i, timesteps, by = 5)
  }
  ## calculate the activity level of the feedback loop for each cell over time
  multi_cell_feedback_activity <- matrix(ncol = n_cells, nrow = timesteps)
  for (j in 1:n_cells){
    cell_feedback_activity <- vector(length = timesteps/5)
    for(i in 1:(timesteps/5)){
      curr_time <- i
      cell_id <- j
      sc_feedback_state <- multi_cell_state_time[c(8, 9, 10, 12),cell_id,time_slices[i,]]
      sum_activity <- sum(sc_feedback_state)/(4*5)
      cell_feedback_activity[i] <- sum_activity
    }
    cell_feedback_activity <- rep(cell_feedback_activity, each = 5)
    multi_cell_feedback_activity[,j] <- cell_feedback_activity
  }
  ## calculate whether the cell has entered further development or not
  mc_chance_of_dev <- matrix(ncol = n_cells, nrow = timesteps)
  active_count <- numeric(length = n_cells)
  for (i in 1:(timesteps)){
    fully_active <- as.numeric(1 == as.numeric(multi_cell_feedback_activity[i,]))
    active_count <- active_count + fully_active
    mc_chance_of_dev[i,] <- active_count
  }
  mc_chance_of_dev <- as.data.frame(apply(mc_chance_of_dev, 1, 2, function(x) min(max(x, 0), dev_needed))/dev_needed)
  mc_chance_of_dev <- cbind(seq(1, timesteps), mc_chance_of_dev) ; colnames(mc_chance_of_dev)[1] <- "time"

  ## calculate the ratio of cells that have entered development
  develop_finished_ratio <- as.data.frame(matrix(ncol = 1, nrow = timesteps))
  develop_finished_ratio <- cbind(seq(1, timesteps), develop_finished_ratio) ;
  colnames(develop_finished_ratio)[1] <- "time"
  for (i in 1:(timesteps)){
    develop_finished_ratio[i,2] <- (sum(1 == mc_chance_of_dev[i, -1]))/n_cells
  }

  ## get the first timepoint where the cell has developed
  development_init <- numeric(length = n_cells)
  for (i in 1:n_cells){
    development_init[i] <- match(1, mc_chance_of_dev[,i+1])
  }
}

```

```

}

# create matrix to hold the coordinates of time slices
time_slices <- matrix(ncol = time_smooth, nrow = timesteps/time_smooth)
for(i in 1:time_smooth){
  time_slices[,i] <- seq(i, timesteps, by = time_smooth)
}
## calculate the average activity level of the feedback loop for each cell over time
multi_cell_feedback_activity <- matrix(ncol = n_cells, nrow = timesteps)
for (j in 1:n_cells){
  cell_feedback_activity <- vector(length = timesteps/time_smooth)
  for(i in 1:(timesteps/time_smooth)){
    curr_time <- i
    cell_id <- j
    sc_feedback_state <- multi_cell_state_time[c(8, 9, 10, 12),cell_id,time_slices[i,]]
    sum_activity <- sum(sc_feedback_state)/(4*time_smooth)
    cell_feedback_activity[i] <- sum_activity
  }
  cell_feedback_activity <- rep(cell_feedback_activity, each = time_smooth)
  multi_cell_feedback_activity[,j] <- cell_feedback_activity
}
# add time point data and format the matrix to a dataframe
mc_fb_df <- as.data.frame(multi_cell_feedback_activity)
mc_fb_df <- cbind(seq(1, timesteps), mc_fb_df)
colnames(mc_fb_df)[1] <- "time"

# get data for signalling molecule levels
PSF_time <- as.data.frame(t(multi_cell_state_time[13,,])); PSF_time <- cbind(seq(1, timesteps), PSF_time);
colnames(PSF_time)[1] <- "time"
CMF_time <- as.data.frame(t(multi_cell_state_time[15,,])); CMF_time <- cbind(seq(1, timesteps), CMF_time);
colnames(CMF_time)[1] <- "time"
BAC_time <- as.data.frame(t(multi_cell_state_time[14,,])); BAC_time <- cbind(seq(1, timesteps), BAC_time);
colnames(BAC_time)[1] <- "time"
cAMP_time <- as.data.frame(t(multi_cell_state_time[16,,])); cAMP_time <- cbind(seq(1, timesteps),
cAMP_time); colnames(cAMP_time)[1] <- "time"

##### PLOT RESULTS #####
print("generating plots")
## plot levels of signalling molecules
signal_plot <- ggplot(PSF_time, aes(x = time)) + ylim(0, 200)
# add the level of PSF
for (i in 1:n_cells){
  signal_plot <- add_line_to_plot(signal_plot, PSF_time, "time", (colnames(PSF_time)[-1])[i], "magenta3",
0.1)
}
cells_to_highlight <- sample(colnames(PSF_time)[-1], size = 5)
for (i in 1:5){
  signal_plot <- add_line_to_plot(signal_plot, PSF_time, "time", cells_to_highlight[i], "magenta4", 0.3)
}
# add the level of CMF
for (i in 1:n_cells){
  signal_plot <- add_line_to_plot(signal_plot, CMF_time, "time", (colnames(PSF_time)[-1])[i], "yellow3",
0.1)
}
cells_to_highlight <- sample(colnames(PSF_time)[-1], size = 5)
for (i in 1:5){
  signal_plot <- add_line_to_plot(signal_plot, CMF_time, "time", cells_to_highlight[i], "yellow4", 0.3)
}
# add the level of BAC
for (i in 1:n_cells){
  signal_plot <- add_line_to_plot(signal_plot, BAC_time, "time", (colnames(BAC_time)[-1])[i],
"dodgerblue1", 0.12)
}
for (i in 1:5){
  signal_plot <- add_line_to_plot(signal_plot, BAC_time, "time", cells_to_highlight[i], "dodgerblue3",
0.5)
}
# add the level of cAMP
for (i in 1:n_cells){
  signal_plot <- add_line_to_plot(signal_plot, cAMP_time, "time", (colnames(BAC_time)[-1])[i], "tomato1",
0.2)
}
for (i in 1:5){
  signal_plot <- add_line_to_plot(signal_plot, cAMP_time, "time", cells_to_highlight[i], "tomato4", 0.6)
}
signal_plot <- signal_plot + labs(title = "signal concentrations", x = "timestep", y = "magenta: PSF and
CMF, blue: bacteria, red: cAMP")

## plot the activity of developmental circuitry
activity_plot <- ggplot(mc_fb_df, aes(x = time))
# add the activity of all cells with low opacity
for (i in 1:n_cells){
  activity_plot <- add_line_to_plot(activity_plot, mc_fb_df, "time", (colnames(mc_fb_df)[-1])[i],
"green3", 0.3)
}
# highlight some of the cells with high opacity
cells_to_highlight <- sample(colnames(mc_fb_df)[-1], size = 5)
for (i in 1:5){
  activity_plot <- add_line_to_plot(activity_plot, mc_fb_df, "time", cells_to_highlight[i], "green4", 0.5)
}
# add the developmental potential of all cells at low opacity
for (i in 1:n_cells){
  activity_plot <- add_line_to_plot(activity_plot, mc_fb_df, "time", (colnames(mc_fb_df)[-1])[i],
"purple1", 0.2)
}
# highlight some of the cells with high opacity

```

```

        for (i in 1:5){
          activity_plot <- add_line_to_plot(activity_plot, mc_chance_of_dev, "time", cells_to_highlight[i],
"purple3", 0.6)
        }
        activity_plot <- activity_plot + geom_line(data = develop_finished_ratio, aes(x = time, y = V1),
color="red", size=2, alpha = 0.9) + labs(title = "developmental activity", x = "timestep", y = "green: ACP, purple:
ACP targets, red: competent cells" )

      ## plot the histogram of development start times
      dev_init_df <- as.data.frame(development_init)
      meanval <- round(mean(development_init), 2)
      sdval <- round(sd(development_init), 2)
      dev_start_plot <- ggplot(dev_init_df, aes(x = development_init)) +
        xlim(0, timesteps) +
        geom_histogram(aes(x = development_init , y = ..density..), fill="black", colour="black", alpha = 0.25,
bins = 25) +
        stat_function(fun = dnorm, args = list(mean = meanval, sd = sdval), col = "#1b98e0", size = 0.8, alpha
= 0.7) +
        labs(title = "time of aggregation competence", x = "timestep", y = "density", subtitle =
paste0(paste0("mean = ", meanval), paste0(" sd = ", sdval)))
      print("rendering plots")
      # combine plots
      ggarrange(signal_plot, activity_plot, dev_start_plot,
                labels = c("A", "B", "C"),
                ncol = 3, nrow = 1)
    }

  # this function calculates statistics for the entire simulation
  multisim_stats<- function(input_multi_cell_state_time){
    # set the number of cycles the cAMP pathway should be active to initiate development
    dev_needed <- 200
    multi_cell_state_time <- input_multi_cell_state_time
    timesteps <- dim(multi_cell_state_time)[3]
    n_cells <- dim(multi_cell_state_time)[2]

    ## calculate the activity level of the feedback loop for each cell over time
    multi_cell_feedback_activity <- matrix(ncol = n_cells, nrow = timesteps)
    for (j in 1:n_cells){
      cell_feedback_activity <- vector(length = timesteps)
      for(i in 1:(timesteps)){
        curr_time <- i
        cell_id <- j
        sc_feedback_state <- multi_cell_state_time[c(8, 9, 10, 12), cell_id, i]
        sum_activity <- sum(sc_feedback_state)/(4)
        cell_feedback_activity[i] <- sum_activity
      }
      multi_cell_feedback_activity[,j] <- cell_feedback_activity
    }

    ## calculate whether the cell has entered further development or not
    mc_chance_of_dev <- matrix(ncol = n_cells, nrow = timesteps)
    active_count <- numeric(length = n_cells)
    for (i in 1:(timesteps)){
      fully_active <- as.numeric(1 == as.numeric(multi_cell_feedback_activity[i,]))
      active_count <- active_count + fully_active
      mc_chance_of_dev[i,] <- active_count
    }
    mc_chance_of_dev <- (apply(mc_chance_of_dev, c(1, 2), function(x) min(max(x,0),dev_needed))/dev_needed)
    mc_chance_of_dev <- cbind(seq(1, timesteps), mc_chance_of_dev)

    ## get the first timepoint where the cell has developed
    development_init <- numeric(length = n_cells)
    for (i in 1:n_cells){
      development_init[i] <- match(1, mc_chance_of_dev[,i])
    }

    # get data of cAMP in last state
    cAMP_time <- (multi_cell_state_time[16,,timesteps])
    camp_mean <- mean(cAMP_time) ; camp_sd <- sd(cAMP_time)

    # calculate dev inititaion time stats
    dev_mean <- mean(development_init); dev_sd <- sd(development_init)

    return(c(camp_mean, camp_sd, dev_mean, dev_sd))
  }

#####
### example of usage
multi_cell_state <- multisim_prep(n_cells = 30,
                                    quorum_starting = 15,
                                    bac_starting = 85,
                                    envir_var = 30,
                                    signal_fail = 1,
                                    signal_fail_var = 10,
                                    cell_fail = 1,
                                    cell_fail_var = 5)
multi_cell_state_time <- multisim_simulate(multi_cell_state,
                                             timesteps = 10000,
                                             PSF_CMF_up = 0.02,
                                             BAC_down = 0.05,
                                             cAMP_prod = 1.001,
                                             signal_var = 2,
                                             diffusion = 0.0005)
multisim_analysis(multi_cell_state_time, 100)

```