# THE MECHANISMS OF CHLAMYDIAL CELL-FORM DEVELOPMENT

A Dissertation

Presented in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

with a

Major in Microbiology, Molecular Biology, and Biochemistry

in the

College of Graduate Studies

University of Idaho

by

Travis Joseph Chiarelli

Approved by:

Major Professor: Scott S. Grieshaber, Ph.D.

Committee Members: Nicole A. Grieshaber, Ph.D.; Anders Omsland, Ph.D.; Scott A. Minnich, Ph.D.; Lee A. Fortunato, Ph.D.

Department Administrator: Tanya A. Miura, Ph.D.

August 2022

THE MECHANISMS OF CHLAMYDIAL CELL-FORM DEVELOPMENT

ABSTRACT

By Travis Joseph Chiarelli
University of Idaho
August 2022

Chair: Scott Stephen Grieshaber

*Chlamydiae* are obligate intracellular Gram-negative bacteria that infect an array of eukaryotic hosts. *Chlamydia trachomatis*, a human-adapted species, is the global leading cause of bacterial sexually transmitted infections as well as trachoma, a preventable form of blindness. All *Chlamydiae* progress through an essential biphasic developmental cycle consisting of two primary cell forms. The elementary body, or EB, is the infectious, non-replicating, cell form. Whereas, the reticulate body, or RB, is non-infectious, but replication competent. The infectious cycle is initiated by the EB via pathogen-mediated endocytosis. Once inside the host, development occurs within a parasitophorous vacuole, termed the inclusion. Within the first 11 hours of infection, the nascent inclusion migrates to the microtubules organization center and the EB undergoes primary differentiation into the replicating RB. At approximately 20 hours post infection, a subset of RBs begins secondary differentiation back into infectious EBs. Secondary differentiation continues through the remainder of the infectious cycle until host cell lysis or inclusion extrusion releases the EBs into the environment to initiate subsequent rounds of infection. Although the ability to transition from EB-to-RB-to-EB is essential for chlamydial growth and proliferation, the mechanisms that regulate *Chlamydia* cell-form development remain largely unknown.

This dissertation demonstrates the power of combining automated live-cell microscopy and cell-form specific reporter strains to monitor chlamydial developmental dynamics in active infections. Computational models were developed to test multiple chlamydial developmental hypotheses, explore cell-form

subpopulation dynamics, and guide *in vivo* experiments. The data from these experiments suggests that *Chlamydia* is not only responding to an intrinsic developmental signal, but that cell-form differentiation is a multi step process consisting of both cell division dependent (RB-to-IB asymmetric production) and independent (IB-to-EB direct maturation) mechanisms. Lastly, to elucidate the genes involved in RB-to-EB differentiation we developed a mutagenesis screen using automated live-cell microscopy and a dual cell-form specific chlamydial reporter strain.

## ACKNOWLEDGMENTS

In these acknowledgements I would like to thank the many people who supported me through this period of my scientific career both professionally and personally.

Scott Grieshaber Ph.D. and Nicole Grieshaber Ph.D.: Thank you. Working with and under you has been a blast! You have been great advisors and mentors. Scott, you supported me in several scientific endeavors, including an array of new-to-me microscopic techniques as well as computer programming (something I am still working on until this day). You have given me an appreciation for combining computer science with bench work that I otherwise would not have. Nicole, you have been essential to my success, providing me with much needed guidance throughout my graduate career; you will be sorely missed as I move on. You both have focused on my future development as a scientist, working closely with me to write several grant proposals and manuscripts. I appreciate all the work you have put in for me, and without the two of you I would not be the scientist I am today. Thank you both for being the best mentor team I could have asked for.

My committee, Anders Omsland Ph.D.; Scott (Sam) Minnich Ph.D.; and Lee Fortunato Ph.D.: Having you as my committee members has been a wonderful experience. Your collective guidance has been greatly appreciated, as has the support you've shown me throughout this process. Sam and Anders, you both emphasized the importance of not only fully understanding my scientific questions but also broadening my knowledge to include adjacent fields of research. These lessons have helped me develop into the scientist I am today. Lee, although you came late to my committee, you have been a great support system, and I have cherished our friendship. Thank you for being an open door and always willing to chat, regardless of the topic. You have helped me stay grounded, especially in these last months of my graduate career, again thank you.

Cody Appa: You have been a vital support system for me. Working with and mentoring you has been a great experience and taught me many things. However, it is our friendship that I truly cherish. Your willingness to step up and help others

(myself included) is a fantastic quality and one I hope you never lose. Thank you for your support both as a friend and colleague.

Cameron Mandel: You have been an awesome friend, colleague, and collaborator. I turn to you often for an outside view on my current thoughts and you always provide me with invaluable feedback. I also wanted to thank you for including me in your research projects; working with you and Anders has emphasized the importance of collaboration, something I will carry with me as I continue my research career.

Biological Sciences Graduate Students: I want to thank many of the other graduate students. The ability to connect with you all throughout my stint in Moscow has been a life saver. A big thank you to Salvador Castaneda and Ian Oiler, you are two of the closest friends I have made in Moscow and I will miss you both dearly. Emmanuel (Manny) Ijezie, since you joined Lee's lab (and are now only a door away), we have become close friends and colleagues. You have been a great support in the final chapter of this process. I have greatly appreciated having you as an accountability buddy and the periodic check-ins we give one another.

Undergraduate researchers, Sarah Gibbs; Grace Neiswanger; Kristina Peretti; Zach Borshe; and Lucas LaMana: Thank you for teaching me how to teach others. Mentoring each of you and watching you all grow and develop as researchers was a truly rewarding experience and has kept me jazzed about research and hopeful of the future.

Noonball team: Thank you to all of the Noonballers. Noonball has been a great support and outlet, both socially and physically. A special thank you to Bill Smith for keeping Noonball up and running, even through the pandemic. It seemed like quite a headache at times, and it did not go unappreciated.

My family and friends: Thank you for all your support and guidance throughout this journey; it was invaluable, and I would not have made it without you. I love you.

Mary Chiarelli: You are my rock, my best friend, my partner. Thank you for all you have done for me. You have provided me with endless love and encouragement. You have been my biggest fan and never doubted my capabilities. You have supported me throughout much of my life and for this I will be forever grateful.

## TABLE OF CONTENTS

## LIST OF TABLES

# LIST OF FIGURES AND MOVIES

## LIST OF SUPPLEMENTAL MATERIALS

CHAPTER ONE: INTRODUCTION

**_Chlamydia trachomatis_: disease and pathogenesis.**

_Chlamydia trachomatis_ is a prominent human pathogen, consisting of 15 distinct serovars [1]. Serovars A-C infect the epithelial cells of the conjunctiva and are the causative agent of ocular trachoma, the most common form of preventable blindness worldwide [2, 3]. The LGV serovars (L1, L2, and L3) cause lymphogranuloma venereum by dissemination into the lymphatic system via infection of macrophages. LGV infections can lead to genital and anorectal ulcers, lymphadenopathy, and fibrotic scarring [4, 5, 6]. The urogenital serovars (D-K) are responsible for the sexually transmitted infection (STI) chlamydia. Chlamydia is the most reported bacterial STI worldwide, with the United States alone reporting 1.8M _C. trachomatis_ infections in 2019 [7, 8]. Sexual transmitted chlamydial infections are often subclinical and when left untreated can lead to pelvic inflammatory disease, ectopic pregnancy, and sterility [9]. STI _C. trachomatis_ infections are also associated with an increased risk of cervical cancer and transmission of HIV [9, 10, 11]. The total direct medical costs of chlamydial infections within the United States are estimated at >$500 million, with infections affecting all racial/ethnic groups and age ranges [8, 12].

**Chlamydial development.**

_Chlamydiae_ are obligate intracellular Gram-negative bacterial parasites of eukaryotic cells [13, 14]. Chlamydial growth and proliferation is dependent on a biphasic developmental cycle consisting of two primary cell forms. The elementary body, or EB, is the infectious cell form. The EB is the smaller of the cell forms (~0.3 μm), is non-replicative, and contains a condensed nucleoid [15, 16]. The EB outer membrane also contains extensive disulfide cross-linking which allows the EB to remain stable in extracellular osmotically unfavorable conditions [17]. Although appearing spore-like in nature, the EB is still metabolically active and requires nutrient uptake (i.e. glucose 6-phosphate, ATP, and amino acids) to maintain infectivity [18]. Infection of the host is initiated by the EB through electrostatic interactions with the host via heparan sulfate glycosaminoglycans [19]. Host cell invasion then occurs via effector-mediated

endocytosis of the EB. Among the effectors responsible for EB uptake is the type three secreted effector TARP (translocated actin-recruiting phosphoprotein); TARP causes host actin cytoskeletal rearrangement which aids in EB engulfment [20]. Once inside the host, chlamydial development occurs in a membrane-bound, parasitophorous vacuole known as the inclusion. The newly formed inclusion circumvents the endosomal and lysosomal pathway by active protein expression from *Chlamydia* [21]. *Chlamydia* also recruits host dynein to the inclusion membrane to traffic the inclusion down microtubules to the MTOC [22]. This localizes the inclusion near the host nucleus and Golgi where it then intercepts sphingomyelin rich exocytic vesicles from the Golgi [23, 24]. Inside the inclusion, the EB matures into the reticulate body, or RB, in a process that takes between 8-11 hours [16, 25, 26]. The RB is the larger of the cell forms (~1.0 µm), is replication competent, and contains a loose nucleoid structure [16, 25, 27]. The RB replicates symmetrically from ~12-24 hpi, at which point intermediate bodies, or IBs (a transitory form that occurs between the RB and EB) and newly infectious EBs are formed, allowing for subsequent rounds of infection after host cell lysis or inclusion extrusion [16, 28, 29].

As *Chlamydia* progresses through biphasic development, it undergoes large transcriptional changes which have been placed into three major temporal categories (early, mid, and late-cycle). Early-cycle gene expression (classified as 1-8 hpi) is associated with EB-to-RB germination and has been shown to include upregulation of the protein chaperone system *groEL/S*, the chlamydial specific genes *euo* and *ihtA*, as well as several genes involved in metabolite translocation and inclusion formation [25, 28, 30, 31] . Mid-cycle expression (~8-24 hpi) is associated with RB replication, and consists of genes involved in an array of cellular processes, including cell division, DNA replication, inclusion membrane modification (*incA*), and Type III secretion [18, 28]. Lastly, late-cycle expression corresponds to secondary differentiation of RBs to EBs and is considered to be anything >24 hpi. Genes upregulated at the stage of infection are consistent with the EB structural form and are involved in DNA compaction (*hctA, hctB*: both encode histone-like proteins), bacterial cell membrane modification (*pmps*: polymorphic outer membrane protein, *omcA/B*: Cysteine-rich OMP), and Type III secretion [25, 28, 32].

**Chlamydial cell-form regulatory mechanisms.**

Although biphasic development is essential for chlamydial survival and proliferation, very few developmental regulatory elements have been elucidated. Due to the inherent genetic intractability of *Chlamydia*, the majority of the regulatory elements in *Chlamydia* have been interrogated using *in vitro* methods and/or surrogate systems. One of the first developmental regulatory elements discovered was the 27 kD protein EUO (early upstream open reading frame) [33, 34]. EUO is a chlamydial specific DNA-binding protein that is expressed as early as 1 hpi and continues to increase in expression until approximately 20 hpi, times which correspond to primary differentiation and RB replication [33, 34]. EUO was found to be a transcriptional repressor that binds to a consensus sequence within both $\sigma^{28}$ (*ltub, scc2, cdsU, hctB, dnaK-P2, pgk, bioY*) and $\sigma^{66}$ (*copB, omcAB*) dependent promoters, inhibiting their expression [34, 35, 36, 37]. EUO has been classified as a master regulator of late genes as many of the genes repressed by EUO are expressed late in infection, a time concurrent with EB formation [25, 32, 37]. However, these transcriptional results are entirely from *in vitro* assays as a *Δeuo* chlamydial mutant has not been identified, and an *euo* over-expression construct has yet to be transformed into *Chlamydia*.

Pgp4 is a plasmid-encoded transcriptional regulator of other plasmid-encoded and chromosomal genes [38]. Pgp4 has also been implicated as a repressor of late gene expression as *in vitro* assays demonstrated the direct binding of Pgp4 with EUO, leading to an enhancement of EUO's repressive abilities [39]. However, across multiple studies, *Δpgp4* mutants have exhibited decreased expression of several genes, including mid-late gene *glgA* [38, 39, 40, 41]. The mechanism of Pgp4 positive regulation on these genes is currently unknown.

HctA and HctB are two histone-like proteins that are expressed late in the developmental cycle (>20 hpi) [25, 28]. Consistent with their predicted functions, HctA and HctB have been shown to cause DNA condensation when overexpressed in *Escherichia coli* [42, 43]. Both HctA and HctB have also been proposed as late stage

regulators of EB development. HctA was theorized to be an H-NS like protein as it was shown to modulate gene expression of topologically-dependent promoters in *E. coli* by reducing negatively supercoiled DNA [44]. Whereas, HctB was shown to have a high affinity to both RNA and DNA and was capable of preventing *in vitro* gene expression at both the transcriptional and translation level [45]. Yet, how or if HctA and HctB regulate expression in *Chlamydia* remains unknown.

Regulation of HctA and HctB has been shown to occur via multiple mechanisms. Both HctA and HctB were found to be post-translationally regulated by an isoprenoid precursor, 2-C-methyl-D-erythritol-2,4-cyclodiphosphate (MEcPP), produced from the non-mevalonate methylerythritol phosphate pathway (MEP). In *E. Coli*, overexpression of the chlamydial IspE (CTL0173), an intermediate MEP enzyme, led to the dissociation of HctA and HctB from chromosomal DNA, relaxing the condensed DNA structure [42, 43]. However, by partial reconstruction of the MEP pathway (IspD, IspE, IspF) *in vitro*, it was discovered that MEcPP was the causative agent of EB nucleoid relaxation and not IspE directly [42]. As overexpression of IpsE leads to DNA decondensation and RBs are known to contain a loose nucleoid structure, it has been proposed that MEcPP may be involved in the EB-to-RB germination process.

In addition to protein-level modulation, *hctA* has also been found to be regulated at the translational level by a small non-coding RNA, IhtA (inhibitor of *hctA* translation) [31]. In *E. coli*, expression of IhtA was shown to prevent HctA-induced DNA condensation [46]. Using structural *in silico* predictions and mutagenesis of IhtA and *hctA*, it was found that IhtA inhibits *hctA* mRNA translation by binding to the 5'UTR and first 6 base pairs of the *hctA* ORF [31, 47]. IhtA was also found to inhibit the translation of an alternative transcript, CTL0322 (*ddbA*), which contains a partially conserved IhtA/*hctA* binding sequence[47]. DdbA is a highly conserved chlamydial-specific protein that has been linked to the long-term maintenance of infectivity in EBs [15, 48].

Lastly, *Chlamydia* contains a single two-component regulatory system (TCRS), CtcB/CtcC, that is phylogenetically related to the AtoS-NtrB/AtoC-NtrC pathways found in other bacteria [49]. TCRS like AtoS/AtoC or NtrB/NtrC are used by bacteria to

respond to an array of external signals including nitrogen, pH, and osmolarity. AtoS/NtrB is the transmembrane histidine kinase component and utilizes an extracellular sensory domain to respond to environmental stimuli. Upon receiving the proper external cue, AtoS/NtrB undergoes autophosphorylation, subsequently activating AtoC/NtrC by phosphotransfer [50, 51]. AtoC/NtrC is the response regulator component and typically contains three domains: regulatory, DNA-binding, and ATP hydrolyzing. Activated AtoC/NtrC interacts directly with $\sigma^{54}$ and hydrolyzes ATP to induce the open conformation in the $\sigma^{54}$ RNAP, allowing for transcription of the $\sigma^{54}$-dependent genes [52]. In *Chlamydia*, *ctcB* and *ctcC* are upregulated at 24 hpi, corresponding to RB-to-EB development [49]. CtcB, unlike its AtoS/NtrB homologs, does not contain an extracellular sensory domain and is predicted to be cytosolic. However, *in vitro*, CtcB has been shown to undergo autophosphorylation and phosphotransfer to CtcC [49]. CtcC, differing from AtoC/NtrC, contains only a regulatory and ATP hydrolyzing domain and is missing the DNA-binding enhancer region. Yet, overexpression of CtcC in *Chlamydia* has shown that CtcC is still capable of inducing the $\sigma^{54}$ regulon, a large set of late EB-associated genes [41].

The regulatory elements above have been primarily described by their direct interactions with the pathways and genes that they regulate. This has provided the chlamydial field with invaluable information for many intermediate control mechanisms involved in cell-form differentiation. However, the mechanisms or signals that control these regulatory elements remain completely unknown.

### *Chlamydia* genetic manipulation.

**Ectopic expression.** A major hurdle in the study of chlamydial biology has been the lack of genetic tools, however over the last decade there have been many advances. The first major breakthrough was in 2011 with the development of a method for stable transformation in *Chlamydia*. This was performed by the ligation of the chlamydial native plasmid (pL2) to the *E. coli* pBR325 vector [53]. The pBR325 portion of this hybrid construct allowed for high concentrations of plasmid to be grown in *E. coli,* while the pL2 portion allowed for plasmid replication and maintenance in *Chlamydia*. Following the successful development of a stable

chlamydial transformation protocol was the creation of an anhydrotetracycline-inducible (aTC) plasmid-based system. This system has allowed for the conditional expression of epitope-tagged genes, enabling the determination of protein localization within *Chlamydia* and, if secreted, the host cell and infectious environment [54, 55, 56]. More recently, an alternative inducible control mechanism was developed by replacing the ribosome-binding site (RBS) of the highly expressed bacteriophage T5 promoter with a theophylline-inducible DNA aptamer, producing a synthetic riboswitch [40]. The aTC (transcriptional) and riboswitch (translational) inducible systems have been combined within a single promoter to create tighter gene regulation and allow for the introduction of potentially toxic genes. Prior to combining these systems, transformation of a plasmid containing an inducible *hctB* cassette resulted in promoter mutations, leading to the abolishment of exogenous *hctB* expression. However, under aTC and riboswitch control, an exogenous copy of *hctB* was successfully transformed into *Chlamydia*. Over-expression of *hctB* in this strain was shown to produce a dramatic decrease in infectious progeny production [40]. A conditional knockdown of the regulatory protein Pgp4 has also been created by replacing the RBS of *pgp4* with the inducible DNA aptamer (producing the native promoter-E-*pgp4* strain: nprom-E-*pgp4*). In the absence of the inducer, the nprom-E-*pgp4* strain phenocopied the *Δpgp4* strain, L2R, with its inability to produce glycogen [40]. When *pgp4* was knocked down, nprom-E-*pgp4* also exhibited an increase in infectious progeny production [40]. Creation of further overexpression and conditional knockdown strains using the aTC and riboswitch inducible systems will be a powerful tool in determining the function of regulatory proteins and the genetic pathways involved in chlamydial development.

**Reporter *Chlamydia*.** The initial chlamydial transformation protocol utilized a reporter cassette containing a green fluorescent protein (GFP) under the control of the meningococcal class I protein promoter to allow for visual confirmation of chlamydial transformants [53]. Since then, implementation of reporter *Chlamydia* has been widely utilized across the chlamydial field. Of note, was the creation of reporters to monitor the developmental cycle. This method incorporated cell form-

specific promoters (*groESL*:RB, *omcAB*:EB) to drive either a GFP or RFP, allowing for qualification of chlamydial development during active infections [57]. As previous assays for monitoring chlamydial development have been destructive in nature and only allowed for snapshots of the developmental cycle (i.e. EM, RNA-seq, microarray, re-infection), the ability to monitor the developmental cycle throughout the entirety of an active infection will be a useful tool in our understanding of chlamydial biology.

**Mutagenesis.** Mutagenesis is a powerful tool to uncover the function of proteins via chromosomal knockouts. Although used regularly in other genetic systems, applicable mutagenesis techniques in *Chlamydia* are still in their infancy. The first reported use of chemical mutagenesis in *Chlamydia* using ethyl methane-sulfonate (EMS) was in 2011, where Kari et al. performed a proof-of-concept study to create and identify mutations within the *trpBA* operon [58]. EMS primarily produces G-T mispairing by guanine alkylation, leading to either synonymous, non-synonymous, or nonsense mutations across the chromosome. Further EMS studies in *Chlamydia* have led to the identification of nonsense mutations in genes involved in numerous biological pathways including glycogen metabolism, glycolysis, DNA mismatch repair, and DNA excision repair [59]. Of note are the EMS mutant clones of *gyrA2*, a DNA gyrase, and *gspE*, a type 2 secretion component, which were identified and linked to defects in chlamydial proliferation and infectious progeny production [60].

A second method of random mutagenesis developed more recently was transposon mutagenesis. This method utilizes chlamydial transformation to introduce a non-stable plasmid containing the C9 Himar1 transposase and transposable element into *Chlamydia*. The advantage of this method is that instead of producing single base pair mutation, as in chemical mutagenesis, which can easily revert, the transposon contains a large disruptive cassette. This cassette can also contain fluorescent reporters or antibiotic resistance genes, allowing for easy screening and selection from the un-mutagenized population. Two studies have implemented this technique in both *C. trachomatis* and *C. muridarum*, producing 105 and 33 mutants, respectively. The identification of CTL0593 in *C. trachomatis*, a gene that encodes a

competence-associated protein homologous to ComEC found in other Gram-negative and Gram-positive bacteria, and the knockout and plasmid complementation of the chlamydial polymorphic outer membrane protein gene, *pmpI,* in *C. muridarum* were both achieved using this method [61, 62]. The major disadvantages of random mutagenesis techniques are that most mutants contain multiple DNA lesions, therefore whole genome sequencing is needed to identify mutant alleles and backcrossing is required to produce isogenic mutants. This is a time-intensive process and the closer mutations are to one another the less likely of a successful backcross. These methods also use large screening pools which makes initial mutant identification laborious.

Since the advent of chlamydial transformation, multiple methods for site-specific gene deletion have been developed. The first technique developed was a modified version of the TargeTron system which uses the insertion of group 2 introns into the chromosome. Although minimally implemented, a proof-of-concept study knocked out *incA*, a gene whose product is responsible for inclusion fusion and produces an obvious multi-inclusion phenotype [63]. The chlamydial anti-anti sigma factor *rsbV* and the *groEL* paralogs, *ChgroEL2* and *ChgroEL3*, have also been disrupted using this system, each demonstrating a decrease in infectious progeny production [64, 65]. A study of intron insertion stability was also performed using a mouse vaginal tract infection model, where integrated introns were shown to be stable for over 27 days [66]. Lastly, TargeTron has been used to knock out the chlamydial inclusion membrane protein CTL0480. The ΔCTL0480 mutant demonstrated a loss of host myosin phosphatase (MYPT1) recruitment to the inclusion membrane and increased rates of inclusion egress [67].

FRAEM (fluorescence-reported allelic exchange mutagenesis), an alternative method to TargeTron, utilizes an inducibly-stable plasmid system and homologous recombination between the plasmid and chromosome to introduce a site-specific cassette containing both a fluorescent reporter and antibiotic resistance gene that disrupts or replaces the loci of interest [68, 69]. In *Chlamydia*, FRAEM has been used to knock out the alpha subunit of the tryptophan synthase, *trpA*, producing a chlamydial mutant incapable of utilizing indole, a tryptophan precursor [68]. Two chlamydial

secreted effectors, *tmeA* and *tmeB* (translocated membrane-associated effectors), have also been knocked out using FRAEM, and although no discernable phenotype was detected in the *ΔtmeB* mutant, the *ΔtmeA* strain exhibited a large decrease in infectivity [69].

Both TargeTron and FRAEM are examples of insertional mutation systems and leave integrated cassettes within the targeted genes. These chromosomal insertions can lead to polar effects in adjacent genes, disrupting their regulation. To overcome this issue, FLAEM (floxed cassette allelic exchange mutagenesis) was developed by the modification of the FRAEM plasmid to include flanking loxP sites around the insertional cassette and the incorporation of a secondary plasmid containing the Cre recombinase for loxP directed cassette excision [70]. As FLAEM is a newer molecular system, only a proof-of-concept study has been performed with the *tmeA* gene. However, excision of the integrated cassette from *tmeA* was able to reverse the polar effects on *tmeB*, returning *tmeB* expression to wildtype levels in the *ΔtmeA-lx* mutant [70, 71].

As *Chlamydia* typically harbors a stable cryptic plasmid, a major advantage of both allelic exchange mutagenesis systems (FRAEM, FLAEM) is the introduction of an inducibly-stable plasmid variant. These systems utilize aTC to induce the expression of *pgp6*, a plasmid-encoded gene associated with plasmid stability [68]. After cassette integration has occurred, aTC is removed and the plasmid is eventually lost upon successive rounds of chlamydial replication, allowing for the introduction of a new plasmid and making trans complementation possible.

Because *Chlamydia* progresses through an essential developmental cycle, genes that are central to development can be difficult to study with above methods as mutant *Chlamydia* may be non-viable. However, in 2018, an inducible CRIPSR interference (CRIPSRi) system was developed in *Chlamydia*, allowing for the conditional knockdown of essential genes [72]. The original study utilized a modified version of the *Staphylococcus aureus* catalytically inactive Cas9 (dCas9) under tetracycline-inducible control. Even though transformation of the dCas9 plasmid was unstable in *Chlamydia*, as a proof-of-concept, reversible repression of *incA* was demonstrated [72]. The CRIPSRi system has since been improved upon and

implemented to study the mid-cycle chlamydial protease system *clpP2X* [72, 73]. This study demonstrated that *Chlamydia* deficient in *clpP2X* exhibited severe bacterial growth defects as well as a reduction in infectious progeny [73].

Although several studies have implemented transformation-based mutagenesis systems, wide use of these techniques has proven difficult. This is likely due to many aspects including large constructs size (>15kb), the low efficiency of chlamydial transformation (especially in mutant strains), and plasmid stability issues [70, 71, 72, 74]. However, as the chlamydial field progresses many of these techniques will likely improve, making genetic manipulation of *Chlamydia* more accessible.

**Dissertation Summary.**

Several intermediate regulatory elements of chlamydial cell-form development have been elucidated, yet the signals and mechanisms that trigger these elements remain unknown. The largest hurdle in understanding chlamydial biology has previously been the lack of genetic tools. The following chapters take advantage of the recent advancements in genetic manipulation to uncover the mechanisms and signals that *Chlamydia* utilizes to progress through development.

Regulation of RB-to-EB development is essential for the chlamydial dissemination and pathogenesis. In the absence of a known developmental control mechanism, several hypotheses have been proposed. These signals include nutrient limitation, replication-dependent RB size reduction, and contact of the RB T3SS to the inclusion membrane [26, 75, 76, 77]. Using mathematical modeling in combination with cell-form specific reporter stains, **Chapters 2** aimed to determine the nature of the signal of RB-to-EB differentiation to which *Chlamydia* is responding. Data from this study showed that the time to EB differentiation was unaffected by competition assays (increased MOI and superinfection), However, chlamydial growth and development corresponded directly to modulations of the infectious environment temperature and host cell protein synthesis inhibition. *Chlamydia* starved of iron, treated with IFN-γ (tryptophan starved), or inhibited in cell division also exhibited dramatically altered developmental kinetics as well as reductions in EB formation. These results suggest that *Chlamydia* is likely responding to a cell autonomous

process linked to bacterial growth and cell division.

Chapter 3 further explores the cell-form specific developmental dynamics of *Chlamydia* at an individual and subpopulation level. Agent-based modeling was employed to determine the role of cell division in RB-to-EB development. The results from this chapter showed that individual RBs are static late in the developmental cycle and do not convert to IBs after inhibition of cell division or DNA replication. Further IB production was also inhibited after cell replication was prevented. IB-to-EB development, however, was shown to continue after cell division inhibition or the induction of RB lysis. Overall, these data suggest that mature RBs ($RB_E$s) are a stem cell-like population that produce IB daughter cells by asymmetric division and that IB-to-EB development occurs post cell division by direct maturation.

In Chapter 4, to determine the genes involved in RB-to-EB development, we created a forward genetic approach to isolate chlamydial mutants with altered developmental profiles by combining chemical mutagenesis of a cell-form specific chlamydial reporter strain with automated live-cell fluorescence microscopy. Future use of this protocol will aid in the identification of the genes involved in cell-form development in *Chlamydia*.

Many paths were taken in an attempt to more fully understand the chlamydial developmental cycle. Chapter 5 is a summation of unpublished results and includes preliminary data on cell-form developmental dynamics throughout the infectious cycle. Lastly, Chapter 6 summarizes the findings, conclusions, and pitfalls of this work, as well as addresses the future directions of study relating to chlamydial cell-form development.

**References**

1. Elwell C, Mirrashidi K, Engel J. Chlamydia cell biology and pathogenesis. *Nature reviews Microbiology*. 2016;14(6). doi:10.1038/nrmicro.2016.30

2. Thylefors B, Ngrel AD, Pararajasegaram R, Dadzie KY. Global data on blindness. *Bulletin of the World Health Organization*. 1993;73(1).

3. Andreasen AA, Burton MJ, Holland MJ, Polley S, Faal N, Mabey DCW, Bailey RL. Chlamydia trachomatis ompA variants in trachoma: what do they tell us? *PLoS neglected tropical diseases*. 2008;2(9). doi:10.1371/journal.pntd.0000306

4. Mabey D, Peeling RW. Lymphogranuloma venereum. *Sexually transmitted infections*. 2002;78(2). doi:10.1136/sti.78.2.90

5. Mpiga P, Ravaoarinoro M. Effects of sustained antibiotic bactericidal treatment on Chlamydia trachomatis-infected epithelial-like cells (HeLa) and monocyte-like cells (THP-1 and U-937). *International journal of antimicrobial agents*. 2006;27(4). doi:10.1016/j.ijantimicag.2005.11.010

6. de Vries HJC, Smelov V, Middelburg JG, Pleijster J, Speksnijder AG, Morr SA. Delayed microbial cure of lymphogranuloma venereum proctitis with doxycycline treatment. *Clinical infectious diseases: an official publication of the Infectious Diseases Society of America*. 2009;48(5). doi:10.1086/597011

7. Global prevalence and incidence of selected curable sexually transmitted diseases: overview and estimates. Published April 29, 2022.

8. National Academies of Sciences E, Medicine;Health, on Population Health MD, on Prevention PHP, in the United States C of STI, Crowley JS, Geller AB, Vermund SH. Sexually Transmitted Infections: Adopting a Sexual Health Paradigm. Published online March 24, 2021.

9. Paavonen J, Eggert-Kruse W. Chlamydia trachomatis: impact on human reproduction. 1999;5(5). doi:10.1093/humupd/5.5.433

10. Anttila T, Saikku P, Koskela P, Bloigu A, Dillner J, Ikheimo I, Jellum E, Lehtinen M, Lenner P, Hakulinen T, Nrvnen A, Pukkala E, Thoresen S, Youngman L, Paavonen J. Serotypes of Chlamydia trachomatis and risk for development of cervical squamous cell carcinoma. *JAMA*. 2001;285(1). doi:10.1001/jama.285.1.47

11. Peterman TA, Newman DR, Maddox L, Schmitt K, Shiver S. Risk for HIV following a diagnosis of syphilis, gonorrhoea or chlamydia: 328,456 women in Florida, 2000-2011. *International journal of STD & AIDS*. 2015;26(2). doi:10.1177/0956462414531243

12. Kumar S, Chesson HW, Spicknall IH, Kreisel KM, Gift TL. The Estimated Lifetime Medical Cost of Chlamydia, Gonorrhea, and Trichomoniasis in the United States, 2018. *Sexually transmitted diseases*. 2021;48(4). doi:10.1097/OLQ.0000000000001357

13. Moulder JW. Interaction of chlamydiae and host cells in vitro. *Microbiological reviews*. 1991;55(1). doi:10.1128/mr.55.1.143-190.1991

14. Essig A, Heinemann M, Simnacher U, Marre R. Infection of Acanthamoeba castellanii by Chlamydia pneumoniae. *Applied and environmental microbiology*. 1997;63(4). doi:10.1128/aem.63.4.1396-1399.1997

15. Grieshaber NA, Runac J, Turner S, Dean M, Appa C, Omsland A, Grieshaber SS. The sRNA Regulated Protein DdbA Is Involved in Development and Maintenance of the Chlamydia trachomatis EB Cell Form. *Frontiers in Cellular and Infection Microbiology*. 2019;11. doi:10.3389/fcimb.2021.692224

16. Friis RR. Interaction of L cells and Chlamydia psittaci: entry of the parasite and host responses to its development. *Journal of bacteriology*. 1972;110(2). doi:10.1128/jb.110.2.706-721.1972

17. Hackstadt T, Caldwell HD. Effect of proteolytic cleavage of surface-exposed proteins on infectivity of Chlamydia trachomatis. *Infection and immunity*. 1985;48(2). doi:10.1128/iai.48.2.546-551.1985

18. Grieshaber S, Grieshaber N, Yang H, Baxter B, Hackstadt T, Omsland A. Impact of Active Metabolism on Chlamydia trachomatis Elementary Body Transcript Profile and Infectivity. *Journal of bacteriology*. 2018;200(14). doi:10.1128/JB.00065-18

19. Su H, Raymond L, Rockey DD, Fischer E, Hackstadt T, Caldwell HD. A recombinant Chlamydia trachomatis major outer membrane protein binds to heparan sulfate receptors on epithelial cells. *Proceedings of the National Academy of Sciences of the United States of America*.

1996;93(20):11143-11148. doi:10.1073/pnas.93.20.11143

20. Clifton DR, Fields KA, Grieshaber SS, Dooley CA, Fischer ER, Mead DJ, Carabeo RA, Hackstadt T. A chlamydial type III translocated protein is tyrosine-phosphorylated at the site of entry and associated with recruitment of actin. *Proceedings of the National Academy of Sciences of the United States of America*. 2004;101(27):10166-10171. doi:10.1073/pnas.0402829101

21. Scidmore MA, Rockey DD, Fischer ER, Heinzen RA, Hackstadt T. Vesicular interactions of the Chlamydia trachomatis inclusion are determined by chlamydial early protein synthesis rather than route of entry. *Infection and immunity*. 1996;64(12). doi:10.1128/iai.64.12.5366-5372.1996

22. Grieshaber SS, Grieshaber NA, Hackstadt T. Chlamydia trachomatis uses host cell dynein to traffic to the microtubule-organizing center in a p50 dynamitin-independent process. *Journal of cell science*. 2003;116(Pt 18). doi:10.1242/jcs.00695

23. Hackstadt T, Scidmore MA, Rockey DD. Lipid metabolism in Chlamydia trachomatis-infected cells: directed trafficking of Golgi-derived sphingolipids to the chlamydial inclusion. *Proceedings of the National Academy of Sciences of the United States of America*. 1995;92(11). doi:10.1073/pnas.92.11.4877

24. Hackstadt T, Rockey DD, Heinzen RA, Scidmore MA. Chlamydia trachomatis interrupts an exocytic pathway to acquire endogenously synthesized sphingomyelin in transit from the Golgi apparatus to the plasma membrane. *The EMBO journal*. 1996;15(5).

25. Belland RJ, Zhong G, Crane DD, Hogan D, Sturdevant D, Sharma J, Beatty WL, Caldwell HD. Genomic transcriptional profiling of the developmental cycle of Chlamydia trachomatis. *Proceedings of the National Academy of Sciences of the United States of America*. 2003;100(14):8478-8483. doi:10.1073/pnas.1331135100

26. Lee JK, Enciso GA, Boassa D, Chander CN, Lou TH, Pairawan SS, Guo MC, Wan FYM, Ellisman MH, Stterlin C, Tan M. Replication-dependent size reduction precedes differentiation in Chlamydia trachomatis. *Nature communications*. 2018;9(1). doi:10.1038/s41467-017-02432-0

27. Matsumoto A, Manire GP. Electron microscopic observations on the effects of penicillin on the morphology of Chlamydia psittaci. *Journal of bacteriology*. 1970;101(1). doi:10.1128/jb.101.1.278-285.1970

28. Shaw EI, Dooley CA, Fischer ER, Scidmore MA, Fields KA, Hackstadt T. Three temporal classes of gene expression during the Chlamydia trachomatis developmental cycle. *Molecular microbiology*. 2000;37(4):913-925. doi:https://doi.org/10.1046/j.1365-2958.2000.02057.x

29. Hybiske K, Stephens RS. Mechanisms of host cell exit by the intracellular bacterium Chlamydia. *Proceedings of the National Academy of Sciences of the United States of America*. 2007;104(27). doi:10.1073/pnas.0703218104

30. Valdivia RH. Chlamydia effector proteins and new insights into chlamydial cellular microbiology. *Current opinion in microbiology*. 2008;11(1). doi:10.1016/j.mib.2008.01.003

31. Grieshaber NA, Grieshaber SS, Fischer ER, Hackstadt T. A small RNA inhibits translation of the histone-like protein Hc1 in Chlamydia trachomatis. *Molecular microbiology*. 2006;59(2):541-550. doi:10.1111/j.1365-2958.2005.04949.x

32. Nicholson TL, Olinger L, Chong K, Schoolnik G, Stephens RS. Global stage-specific gene regulation during the developmental cycle of Chlamydia trachomatis. *Journal of bacteriology*. 2003;185(10). doi:10.1128/JB.185.10.3179-3189.2003

33. Wichlan DG, Hatch TP. Identification of an early-stage gene of Chlamydia psittaci 6BC. *Journal of bacteriology*. 1993;175(10). doi:10.1128/jb.175.10.2936-2942.1993

34. Zhang L, Douglas AL, Hatch TP. Characterization of a Chlamydia psittaci DNA binding protein (EUO) synthesized during the early and middle phases of the developmental cycle. *Infection and immunity*. 1998;66(3). doi:10.1128/IAI.66.3.1167-1173.1998

35. Zhang L, Howe MM, Hatch TP. Characterization of in vitro DNA binding sites of the EUO protein of Chlamydia psittaci. *Infection and immunity*. 2000;68(3). doi:10.1128/IAI.68.3.1337-1349.2000

36. Rosario CJ, Tan M. The early gene product EUO is a transcriptional repressor

that selectively regulates promoters of Chlamydia late genes. *Molecular microbiology*. 2012;84(6):1097-1107. doi:10.1111/j.1365-2958.2012.08077.x

37. Rosario CJ, Hanson BR, Tan M. The transcriptional repressor EUO regulates both subsets of Chlamydia late genes. *Molecular microbiology*. 2014;94(4). doi:10.1111/mmi.12804

38. Song L, Carlson JH, Whitmire WM, Kari L, Virtaneva K, Sturdevant DE, Watkins H, Zhou B, Sturdevant GL, Porcella SF, McClarty G, Caldwell HD. Chlamydia trachomatis plasmid-encoded Pgp4 is a transcriptional regulator of virulence-associated genes. *Infection and immunity*. 2013;81(3). doi:10.1128/IAI.01305-12

39. Zhang Q, Rosario CJ, Sheehan LM, Rizvi SM, Brothwell JA, He C, Tan M. The Repressor Function of the Chlamydia Late Regulator EUO Is Enhanced by the Plasmid-Encoded Protein Pgp4. *Journal of bacteriology*. 2020;202(8). doi:10.1128/JB.00793-19

40. Grieshaber NA, Chiarelli TJ, Appa CR, Neiswanger G, Peretti K, Grieshaber SS. Translational gene expression control in Chlamydia trachomatis. *PloS one*. 2022;17(1). doi:10.1371/journal.pone.0257259

41. Soules KR, LaBrie SD, May BH, Hefty PS. Sigma 54-Regulated Transcription Is Associated with Membrane Reorganization and Type III Secretion Effectors during Conversion to Infectious Forms of Chlamydia trachomatis. *mBio*. 2020;11(5). doi:10.1128/mBio.01725-20

42. Grieshaber NA, Fischer ER, Mead DJ, Dooley CA, Hackstadt T. Chlamydial histone-DNA interactions are disrupted by a metabolite in the methylerythritol phosphate pathway of isoprenoid biosynthesis. *Proceedings of the National Academy of Sciences of the United States of America*. 2004;101(19). doi:10.1073/pnas.0400754101

43. Grieshaber NA, Sager JB, Dooley CA, Hayes SF, Hackstadt T. Regulation of the Chlamydia trachomatis histone H1-like protein Hc2 is IspE dependent and IhtA independent. *Journal of Bacteriology*. 2006;188(14). doi:10.1128/JB.00526-06

44. Barry CE, Brickman TJ, Hackstadt T. Hc1-mediated effects on DNA structure: a potential regulator of chlamydial development. *Molecular microbiology*. 1993;9(2):273-283. doi:10.1111/j.1365-2958.1993.tb01689.x

45. Pedersen LB, Birkelund S, Christiansen G. Purification of recombinant Chlamydia trachomatis histone H1-like protein Hc2, and comparative functional analysis of Hc2 and Hc1. *Molecular microbiology*. 1996;20(2). doi:10.1111/j.1365-2958.1996.tb02618.x

46. Tattersall J, Rao GV, Runac J, Hackstadt T, Grieshaber SS, Grieshaber NA. Translation inhibition of the developmental cycle protein HctA by the small RNA IhtA is conserved across Chlamydia. *PloS one*. 2011;7(10). doi:10.1371/journal.pone.0047439

47. Grieshaber NA, Tattersall JS, Liguori J, Lipat JN, Runac J, Grieshaber SS. Identification of the base-pairing requirements for repression of hctA translation by the small RNA IhtA leads to the discovery of a new mRNA target in Chlamydia trachomatis. *PloS one*. 2015;10(3). doi:10.1371/journal.pone.0116593

48. Hackstadt T, Brickman TJ, 3rd CEB, Sager J. Diversity in the Chlamydia trachomatis histone homologue Hc2. *Gene*. 1993;132(1). doi:10.1016/0378-1119(93)90526-9

49. Koo IC, Stephens RS. A developmentally regulated two-component signal transduction system in Chlamydia. *The Journal of biological chemistry*. 2003;278(19):17314-17319. doi:10.1074/jbc.M212170200

50. Reitzer LJ, Magasanik B. Expression of glnA in Escherichia coli is regulated at tandem promoters. *Proceedings of the National Academy of Sciences of the United States of America*. 1985;82(7). doi:10.1073/pnas.82.7.1979

51. Liolíou EE, Mimitou EP, Grigoroudis AI, Panagiotidis CH, Panagiotidis CA, Kyriakidis DA. Phosphorylation activity of the response regulator of the two-component signal transduction system AtoS-AtoC in E. coli. *Biochimica et biophysica acta*. 2005;1725(3). doi:10.1016/j.bbagen.2005.06.019

52. Schumacher J, Joly N, Rappas M, Zhang X, Buck M. Structures and organisation of AAA  enhancer binding proteins in transcriptional activation. *Journal of structural biology*. 2006;156(1). doi:10.1016/j.jsb.2006.01.006

53. Wang Y, Kahane S, Cutcliffe LT, Skilton RJ, Lambden PR, Clarke IN. Development of a transformation system for Chlamydia trachomatis: restoration

of glycogen biosynthesis by acquisition of a plasmid shuttle vector. *PLoS Pathogens*. 2011;7(9). doi:10.1371/journal.ppat.1002258

54. Wickstrum J, Sammons LR, Restivo KN, Hefty PS. Conditional gene expression in Chlamydia trachomatis using the tet system. *PloS one*. 2013;8(10). doi:10.1371/journal.pone.0076743

55. Rucks EA, Olson MG, Jorgenson LM, Srinivasan RR, Ouellette SP. Development of a Proximity Labeling System to Map the Chlamydia trachomatis Inclusion Membrane. *Frontiers in cellular and infection microbiology*. 2017;7. doi:10.3389/fcimb.2017.00040

56. Olson-Wood MG, Jorgenson LM, Ouellette SP, Rucks EA. Inclusion Membrane Growth and Composition Are Altered by Overexpression of Specific Inclusion Membrane Proteins in Chlamydia trachomatis L2. *Infection and immunity*. 2021;89(7). doi:10.1128/IAI.00094-21

57. Cortina ME, Ende RJ, Bishop RC, Bayne C, Derr I. Chlamydia trachomatis and Chlamydia muridarum spectinomycin resistant vectors and a transcriptional fluorescent reporter to monitor conversion from replicative to infectious bacteria. *PloS one*. 2019;14(6). doi:10.1371/journal.pone.0217753

58. Kari L, Goheen MM, Randall LB, Taylor LD, Carlson JH, Whitmire WM, Virok D, Rajaram K, Endresz V, McClarty G, Nelson DE, Caldwell HD. Generation of targeted Chlamydia trachomatis null mutants. *Proceedings of the National Academy of Sciences of the United States of America*. 2011;108(17). doi:10.1073/pnas.1102229108

59. Kokes M, Dunn JD, Granek JA, Nguyen BD, Barker JR, Valdivia RH, Bastidas RJ. Integrating chemical mutagenesis and whole-genome sequencing as a platform for forward and reverse genetic analysis of Chlamydia. *Cell host & microbe*. 2015;17(5). doi:10.1016/j.chom.2015.03.014

60. Nguyen BD, Valdivia RH. Virulence determinants in the obligate intracellular pathogen Chlamydia trachomatis revealed by forward genetic approaches. *Proceedings of the National Academy of Sciences of the United States of America*. 2012;109(4):1263-1268. doi:10.1073/pnas.1117884109

61. Wang Y, LaBrie SD, Carrell SJ, Suchland RJ, Dimond ZE, Kwong F, Rockey DD,

Hefty PS, Hybiske K. Development of Transposon Mutagenesis for Chlamydia muridarum. *Journal of bacteriology*. 2019;201(23). doi:10.1128/JB.00366-19

62. LaBrie SD, Dimond ZE, Harrison KS, Baid S, Wickstrum J, Suchland RJ, Hefty PS. Transposon Mutagenesis in Chlamydia trachomatis Identifies CT339 as a ComEC Homolog Important for DNA Uptake and Lateral Gene Transfer. *mBio*. 2019;10(4). doi:10.1128/mBio.01343-19

63. Johnson CM, Fisher DJ. Site-specific, insertional inactivation of incA in Chlamydia trachomatis using a group II intron. *PloS one*. 2013;8(12). doi:10.1371/journal.pone.0083989

64. Thompson CC, Griffiths C, Nicod SS, Lowden NM, Wigneshweraraj S, Fisher DJ, McClure MO. The Rsb Phosphoregulatory Network Controls Availability of the Primary Sigma Factor in Chlamydia trachomatis and Influences the Kinetics of Growth and Development. *PLoS pathogens*. 2015;11(8). doi:10.1371/journal.ppat.1005125

65. Illingworth M, Hooppaw AJ, Ruan L, Fisher DJ, Chen L. Biochemical and Genetic Analysis of the Chlamydia GroEL Chaperonins. *Journal of bacteriology*. 2017;199(12). doi:10.1128/JB.00844-16

66. Lowden NM, Yeruva L, Johnson CM, Bowlin AK, Fisher DJ. Use of aminoglycoside $_{32}$ adenyltransferase as a selection marker for Chlamydia trachomatis intron-mutagenesis and in vivo intron stability. *BMC research notes*. 2015;8. doi:10.1186/s13104-015-1542-9

67. Shaw JH, Key CE, Snider TA, Sah P, Shaw EI, Fisher DJ, Lutter EI. Genetic Inactivation of Chlamydia trachomatis Inclusion Membrane Protein CT228 Alters MYPT1 Recruitment, Extrusion Production, and Longevity of Infection. *Frontiers in cellular and infection microbiology*. 2018;8. doi:10.3389/fcimb.2018.00415

68. Mueller KE, Wolf K, Fields KA. Gene Deletion by Fluorescence-Reported Allelic Exchange Mutagenesis in Chlamydia trachomatis. *mBio*. 2016;7(1):e01817-e01815. doi:10.1128/mBio.01817-15

69. McKuen MJ, Mueller KE, Bae YS, Fields KA. Fluorescence-Reported Allelic Exchange Mutagenesis Reveals a Role for Chlamydia trachomatis TmeA in Invasion That Is Independent of Host AHNAK. *Infection and immunity*.

2017;85(12). doi:10.1128/IAI.00640-17

70. Keb G, Hayman R, Fields KA. Floxed-Cassette Allelic Exchange Mutagenesis Enables Markerless Gene Deletion in Chlamydia trachomatis and Can Reverse Cassette-Induced Polar Effects. *Journal of Bacteriology*. 2018;200(24). doi:10.1128/JB.00479-18

71. Keb G, Fields KA. Markerless Gene Deletion by Floxed Cassette Allelic Exchange Mutagenesis in Chlamydia trachomatis. *Journal of visualized experiments: JoVE*. 2020;(155). doi:10.3791/60848

72. Wood NA, Chung KY, Blocker AM, de Almeida NR, Conda-Sheridan M, Fisher DJ, Ouellette SP. Initial Characterization of the Two ClpP Paralogs of Chlamydia trachomatis Suggests Unique Functionality for Each. *Journal of bacteriology*. 2018;201(2). doi:10.1128/JB.00635-18

73. Wood NA, Blocker AM, Seleem MA, Conda-Sheridan M, Fisher DJ, Ouellette SP. The ClpX and ClpP2 Orthologs of Chlamydia trachomatis Perform Discrete and Essential Functions in Organism Growth and Development. *mBio*. 2020;11(5). doi:10.1128/mBio.02016-20

74. Ouellette SP, Blay EA, Hatch ND, Fisher-Marvin LA. CRISPR Interference To Inducibly Repress Gene Expression in Chlamydia trachomatis. *Infection and Immunity*. 2021;89(7). doi:10.1128/IAI.00108-21

75. Omsland A, Sager J, Nair V, Sturdevant DE, Hackstadt T. Developmental stage-specific metabolic and transcriptional activity of Chlamydia trachomatis in an axenic medium. *Proceedings of the National Academy of Sciences of the United States of America*. 2012;109(48):19781-19785. doi:10.1073/pnas.1212831109

76. Peters J, Wilson DP, Myers G, Timms P, Bavoil PM. Type III secretion  la Chlamydia. *Trends in microbiology*. 2007;15(6). doi:10.1016/j.tim.2007.04.005

77. Hoare A, Timms P, Bavoil PM, Wilson DP. Spatial constraints within the chlamydial host cell inclusion predict interrupted development and persistence. *BMC microbiology*. 2008;8. doi:10.1186/1471-2180-8-5

# CHAPTER TWO: SINGLE-INCLUSION KINETICS OF CHLAMYDIA TRACHOMATIS DEVELOPMENT*

Travis J. Chiarelli,[a] Nicole A. Grieshaber,[a] Anders Omsland,[b] Christopher H. Remien,[a] Scott S. Grieshaber[a]

[a]Department of Biological Sciences, University of Idaho, Moscow, Idaho, USA
[b]Paul G. Allen School for Global Animal Health, College of Veterinary Medicine, Washington State University, Pullman, Washington, USA

**Abstract**

The obligate intracellular bacterial pathogen *Chlamydia trachomatis* is reliant on a developmental cycle consisting of two cell forms, termed the elementary body (EB) and the reticulate body (RB). The EB is infectious and utilizes a type III secretion system and preformed effector proteins during invasion, but it does not replicate. The RB replicates in the host cell but is noninfectious. This developmental cycle is central to chlamydial pathogenesis. In this study, we developed mathematical models of the developmental cycle that account for potential factors influencing RB-to-EB cell type switching during infection. Our models predicted that two categories of regulatory signals for RB-to-EB development could be differentiated experimentally, an "intrinsic" cell-autonomous program inherent to each RB and an "extrinsic" environmental signal to which RBs respond. To experimentally differentiate between mechanisms, we tracked the expression of *C. trachomatis* development specific promoters in individual inclusions using fluorescent reporters and live-cell imaging. These experiments indicated that EB production was not influenced by increased multiplicity of infection or by superinfection, suggesting the cycle follows an intrinsic program that is not directly controlled by environmental factors. Additionally, live-cell imaging revealed that EB development is a multistep process linked to RB growth rate and cell division. The formation of EBs followed a progression with expression from the *euo* and *ihtA* promoters evident in RBs, while expression from the promoter for *hctA* was apparent in early EBs/IBs. Finally, expression from the promoters for the true late genes, *hctB, scc2,* and *tarp*, was evident in the maturing EB.

**Importance**

*Chlamydia trachomatis* is an obligate intracellular bacterium that can cause trachoma, cervicitis, urethritis, salpingitis, and pelvic inflammatory disease. To establish infection in host cells, *Chlamydia* must complete a multiple-cell-type developmental cycle. The developmental cycle consists of specialized cells, the EB cell, which mediates infection of new host cells, and the RB cell, which replicates and eventually produces more EB cells to mediate the next round of infection. By developing and testing mathematical models to discriminate between two competing hypotheses for the nature of the signal controlling RB-to-EB cell type switching, we demonstrate that RB-to-EB development follows a cell-autonomous program that does not respond to environmental cues. Additionally, we show that RB-to-EB development is a function of chlamydial growth and division. This study serves to further our understanding of the chlamydial developmental cycle that is central to the bacterium's pathogenesis.

**Introduction**

Chlamydiae are bacterial pathogens responsible for a wide range of diseases in both animal and human hosts [1]. Chlamydia trachomatis, a human-adapted pathogen, comprises over 15 distinct serovars causing both trachoma, the leading cause of preventable blindness, and sexually acquired infections [2]. According to the CDC, C. trachomatis is the most frequently reported sexually transmitted infection in the United States, costing the American health care system nearly $2.4 billion annually [3, 4]. These infections are widespread among all age groups and ethnic demographics, infecting 3% of the human population worldwide [5]. In women, untreated genital infections can result in pelvic inflammatory disease, ectopic pregnancy, and infertility [6–8]. Every year, there are over 4 million new cases of C. trachomatis sexually transmitted infections in the United States[6, 9] and an estimated 92 million cases worldwide [2, 10].

Chlamydia-related disease is entirely dependent on the establishment and maintenance of the pathogen's unique intracellular niche, the chlamydial inclusion, where the bacteria replicate and undergo a biphasic developmental cycle. This cycle generates two unique developmental cell forms: the elementary body (EB) and the reticulate body (RB). The EB cell type mediates host cell invasion via pathogen-mediated endocytosis, while the RB cell type is replication competent but cannot initiate host cell infection [11]. For C. trachomatis serovar L2, the cycle begins when the EB binds to a host cell and initiates uptake through the secretion of effector proteins by a type III secretion system [12]. During entry, the EB is engulfed by the host cell plasma membrane, forming the inclusion vacuole that is actively modified by Chlamydia to block interaction with the host endocytic/lysosomal pathway [13]. The inclusion continues to mature as the EB cell form transitions to the RB cell form. The time from host cell contact to the formation of the mature inclusion containing replication-competent RBs is 11 h [14]. The formation of infectious EB cells occurs reliably between 18 and 20 h postinfection (hpi) [15]. Regulatory control of the transition between the RB and EB is critical for the chlamydial life cycle, as Chlamydia must balance replication versus production of infectious progeny. How Chlamydia regulates this process is currently unclear, although there have been

multiple hypotheses proposed to explain the control of the developmental cycle. Regulatory mechanisms, such as RB access to or competition for inclusion membrane contact [16], reduction in RB size [14], or responses to changes in nutrient availability [17], all have been proposed to control or influence RB-to-EB cell switching.

In this study, we used mathematical modeling to guide experiments to distinguish between factors that influence RB-to-EB development. The chlamydial life cycle was modeled using systems of differential equations. Each model was tested under simulated conditions that indicated that extrinsic versus intrinsic control of EB development could be distinguished experimentally. To test the model predictions, a live-cell imaging system in combination with promoter-reporter constructs was developed to monitor the developmental cycle in real time at the single-inclusion level. We show that neither the limiting membrane hypothesis nor the intra-inclusion nutrient-limiting hypothesis are consistent with our experimental results and that EB development likely follows a cell-autonomous program. Additionally, we show that this intrinsic program is dependent on RB growth and cell division.

## Results

**Modeling chlamydial development.** We developed two mathematical models that represent potential driving forces in promoting EB development. Each model is a system of ordinary differential equations (ODE) that tracks RBs, intermediate bodies (IBs), and EBs over time (**Fig. 2.1**; also see **Supplemental Material 2.S1**). In these models, the development of the EB is controlled by an inhibitory signal that is intrinsic to each bacterium or is environmental, i.e., shared between the bacteria (**Fig. 2.1A** and **B**). The nature of the signal was not specified beyond an inhibitory effect on EB production at high concentrations and its consumption by RBs. The regulatory nature of this signal could be either positive, as in quorum sensing, or negative, such as nutrient limitation. For our simplified model system, we implemented a negative regulator, but the model will generate identical outputs if the regulator is positive in nature. For each of the two models, the signal is consumed by the bacteria over time, and, once depleted, RB-to-EB conversion commences. The models differ in whether all the RBs in the inclusion compete for one pool of this

signal or whether each RB contains an independent internal pool of the inhibitory signal. The output of both models mimics the general kinetics of the chlamydial developmental cycle. Both models produced identical outputs when a multiplicity of infection (MOI) of 1 was simulated (**Fig. 2.1C**). When a change in the replication rate of *Chlamydia* was simulated, the two models again responded similarly, showing that an increased replication rate led to earlier EB production, while a decreased replication rate resulted in delayed EB production (**Fig. 2.1D**). However, the models produced dramatic kinetic differences with a simulated increase in MOI or time-delayed superinfection. Both simulated conditions caused EB formation to occur sooner in the environment-based signal model but had no effect on EB production when modeled with an intrinsic signal (**Fig. 2.1E**). These data indicate that it is possible to experimentally differentiate between whether an environmental signal or an intrinsic program triggers EB development.

**Development of a live-cell reporter system to monitor the chlamydial developmental cycle.** To experimentally differentiate between mechanisms of differentiation based on the response to an environmental or intrinsic signal, we developed a live-cell imaging system using promoter constructs to monitor the chlamydial developmental cycle. The reporter constructs were designed using the promoters of chlamydial genes that are differentially regulated between the RB and EB forms [18]. To generate an RB reporter, the promoter of *ihtA* was used to drive enhanced green fluorescent protein (EGFP) expression. The sRNA IhtA is expressed early upon infection and negatively regulates the EB-specific gene *hctA* [19]. To generate an EB reporter, the promoter and first 30 nucleotides (nt) of the late gene *hctA* were used to drive the expression of the GFP variant Clover. HctA is a small histone-like protein that is involved in the condensation of the chlamydial genome to form the compact nucleoid characteristic of the EB [20]. The upstream promoter region as well as the first 10 codons of the open reading frame (ORF) of *hctA* were used to construct this reporter, as the regulation of HctA expression involves both the promoter and the IhtA binding site contained in the beginning of the ORF [21]. Each reporter was transformed into *C. trachomatis*, generating the strains

*Ctr-ihtA*prom-EGFP and *Ctr-hctA*prom-Clover (see Table 3.1 in the supplemental material). The chlamydial transformants were used to track the developmental cycle of each strain using live-cell time-lapse microscopy and particle tracking to quantify the fluorescent expression of individual inclusions over time [22]. This technique allows for the tracking of gene expression in multiple individual inclusions over the entire developmental cycle while avoiding the inherent variability of whole-population studies on an asynchronous infection. A detailed description of the system is described in our recently published paper [23]. To verify that the fluorescent reporters accurately reflected the developmental cycle, total chlamydial growth was determined by measuring genomic copies by quantitative PCR (qPCR) and EB production by a replating assay to quantify inclusion-forming units (IFU). EGFP expression from the *ihtA* promoter was first detected at 10 hpi and started to level off at 28 hpi (**Fig. 2.2A**). The initial expression from the *ihtA* promoter was in good agreement with the initiation of RB genomic replication, as demonstrated by genome copies (**Fig. 2.2A**). The initiation of RB replication signals the end of the EB-to-RB transition after cell entry. Imaging of the *hctA* promoter-reporter revealed that the Clover signal could be detected first at 18 hpi (**Fig. 2.2B**). Again, these data were in good agreement with the production of infectious progeny, as EBs were first detected at 20 hpi (**Fig. 2.2B**). We measured 50 individual inclusions per strain and found very little interinclusion variability in the timing of the initiation of expression (**Fig. 2.2C** and **D**). This uniformity in developmental timing can be appreciated in a live-cell time-lapse movie of *Ctr-hctA*prom-Clover infections (**Movie 2.S1**). The close agreement between classic methods for monitoring the chlamydial developmental cycle (IFU and genome copies) and the single-inclusion-based fluorescent reporter system described here demonstrates that this system accurately reflects the developmental cycle.

**Chlamydial development is growth rate dependent.** Both models predicted that changes in growth rate would be reflected in EB production kinetics (**Fig. 2.1D**). There is generally a linear relationship between temperature and the square root of growth rate in bacteria [24]. Therefore, to validate the predictions of our two models,

we monitored *Ctr-ihtA*prom-EGFP and *Ctr-hctA*prom-Clover at three temperatures, 35°C, 37°C (control), and 40°C. As expected, at the lower temperature of 35°C, the EB-to-RB lag time increased dramatically and *ihtA*prom-EGFP expression increased more slowly than that of the 37°C control (**Fig. 2.3A**). The lower replication rate at 35°C was also reflected in measured genome copies (**Fig. 2.3B**). Conversely, the lag time to fluorescence detection was reduced and fluorescence increased faster than the control when grown at 40°C (**Fig. 2.3A**). As predicted by our models, time to EB production was also shifted by changes in growth rate, as *hctA*prom-Clover expression began earlier at 40°C and was delayed at 35°C (**Fig. 2.3C**). These results were verified by measuring the production of infectious progeny (**Fig. 2.3D**) and are consistent with previously published literature where *Chlamydia* growth at 33°C was slowed in both inclusion and EB development [25]. Taken together, these data provide strong evidence that the cycle is growth rate dependent and that our experimental system accurately detected changes in chlamydial development.

**EB development is controlled by intrinsic factors and not environmental factors.** The two mathematical models differ principally in the source of the EB development signal: internal versus environmental. The models produced divergent outcomes under conditions where bacteria are competing for a host cell or an intra-inclusion signal versus a signal internal to each RB. Simulations predicted that the time to EB production would be measurably affected by increasing the MOI if the signal was environmental (competitively consumed) but would be unchanged if the signal was intrinsic (internal to each RB) (**Fig. 2.1E**). To more accurately assay EB development by live-cell imaging, two additional EB gene reporters were constructed. The promoters and first 30 nt of *hctB* and *scc2* were inserted upstream of Clover and transformed into *C. trachomatis*, creating *Ctr-hctB*prom-Clover and *Ctr-scc2*prom-Clover, respectively. Like HctA, HctB is a small histone-like protein that is involved in EB nucleoid formation [26], while Scc2 is a chaperone for type III secretion effector proteins [27]. Our published transcriptome sequencing (RNA-seq) data showed that the transcripts for *hctB* and *scc2* were expressed late, corresponding to the timing of EB production [18]. Monolayers were infected with each

of the four strains, with MOIs ranging from 1 to 32 infectious EBs per host cell, and imaged every 30 min for 40 h. The MOI was calculated by infection with a 2-fold dilution series and back calculating from an observed MOI of 1. The fluorescent signals were normalized by MOI, as this more closely represents fluorescence per RB. Expression initiation of the RB reporter *ihtA*prom, and the EB reporters *hctA*prom, *hctB*prom, and *scc2*prom, did not vary as a function of MOI (**Fig. 2.4A** to **D**). The lack of MOI response for the expression of EB genes corresponded closely with EB production as measured by a reinfection assay (**Fig. 2.4E**). Of note is the dramatic difference in the timing of expression between the late genes. *hctA*prom-Clover expression was initiated at 18 h postinfection, while *hctB*prom-Clover and *scc2*prom-Clover expression was initiated 3 h later at 21 hpi.

Our models predicted that both MOI and superinfection would aid in differentiating between cell-autonomous and environmentally influenced development (**Fig. 2.1E**). The MOI data suggested that RB-to-EB developmental switching is not influenced by the host intracellular or the intrainclusion environment but rather is triggered by a signal intrinsic to *C. trachomatis*. To further differentiate between these possibilities, we measured RB and EB gene expression under superinfection conditions. The chlamydial inclusion is derived from the plasma membrane, and interaction with the endocytic membrane system is actively blocked by *Chlamydia* [13]. When multiple EBs infect a cell, they each create individual inclusions that traffic to the microtubule-organizing center (MTOC) of the host cell [28]. This trafficking, along with the expression of IncA, a protein that promotes fusion of individual inclusions, culminates in homotypic inclusion fusion, resulting in a single chlamydial inclusion per host cell [29, 30]. Our environmental signal model predicted that the developmental cycle of *Chlamydia* under superinfection conditions would be dramatically altered (decreased time to EB production) as a function of the developmental stage of the first infection. To test this, cells were infected with unlabeled *C. trachomatis* L2 for 6, 12, and 18 h prior to a second infection with the indicated *C. trachomatis* L2 reporter strains and imaged starting at 9 h after secondary infection (**Fig. 2.5**). Fluorescent signals were measured for inclusions that were verified to be superinfected by imaging for both differential interference

contrast (DIC) and fluorescence, i.e., inclusions containing both labeled and unlabeled *Chlamydia* (**Fig. 2.5A**). Superinfection at any time after initial infection had no effect on the initiation of expression of either *ihtA*prom-EGFP or *hctA*prom-Clover (**Fig. 2.5B** and **C**). The lack of effect on late gene expression was verified with two other late promoter-reporter strains, *Ctr-hctB*prom-Clover and *Ctr-scc2*prom-Clover, 12 h post superinfection (**Fig. 2.5D** and **E**). We verified that superinfection had no effect on the initial production of infectious progeny by performing a replating assay in the presence of spectinomycin (**Fig. 2.5F**).

To further examine any effect of the intrainclusion environment versus the host intracellular environment, we took advantage of a *Chlamydia* mutant that does not express IncA and, therefore, is defective in homotypic inclusion fusion [29]. Cells were pre-infected with an isogenic mutant pair, either *C. trachomatis* J (incA positive and fusogenic [31]) or *C. trachomatis* Js (incA negative and nonfusogenic [31]) for 18 h, and then were superinfected with *Ctr-ihtA*prom-EGFP or *Ctr-hctA*prom-Clover and imaged starting at 9 h post superinfection (**Fig. 2.5G**). Again, there was no apparent change in kinetics between infection alone (no superinfection), superinfection with inclusion fusion, or superinfection without fusion (**Fig. 2.5H** and **I**). Taken together, these data suggest that the timing of RB-to-EB development is an intrinsic pre-programmed property of *Chlamydia* and does not respond to environmental signals.

**Chlamydial cell division is required for EB development.** Time to EB development responded to RB growth rate, suggesting that chlamydial cell division is critical for development (**Fig. 2.3**). To test the role of cell division in EB development, RB replication was halted by treating infected cells with penicillin G (Pen). *C. trachomatis* does not use peptidoglycan as a structural sacculus and does not contain a peptidoglycan cell wall. Instead, peptidoglycan aids cell septation by forming a ring at the cleavage furrow [32]. Therefore, Pen treatment blocks cell septation but not cell growth.

To assess the effects of Pen treatment on chlamydial developmental kinetics, an additional early gene promoter-reporter, *euo*prom-Clover, was constructed. EUO (early upstream ORF) is a transcriptional repressor that selectively regulates

promoters of *C. trachomatis* late genes and was highly expressed in our RNA-seq data set [18, 33]. Cells infected with *Ctr-euo*prom-Clover or *Ctr-hctA*prom-Clover were treated with Pen at 14 hpi and imaged for a further 34 h (**Fig. 2.6**). The *euo*prom-Clover signal after Pen treatment continued to increase, as did the size of the aberrant RB cells (**Fig. 2.6B** and **7A** and **B**). The expression of *euo*prom-Clover in the presence of Pen also matched the increase in genome copies, which, as previously reported [34], was also Pen insensitive (**Fig. 2.6D**). Unlike *euo*prom-Clover expression, the *hctA*prom-Clover signal was dramatically affected by Pen treatment (**Fig. 2.6B**). The expression of *hctA*prom-Clover was initially repressed by Pen treatment at 14 hpi compared to that of untreated samples; however, expression was initiated 9 h after treatment. We explored this late gene expression behavior further using three other late gene promoter strains, *Ctr-hctB*prom-Clover, *Ctr-scc2*prom-Clover, and *Ctr-tarp*prom-Clover (**Fig. 2.6B** and **Fig. 2.S1**). The Clover expression patterns driven by *hctB*prom, *scc2*prom, and *tarp*prom were dramatically different from that of *hctA*prom, as none showed Clover expression in the Pen-treated samples (**Fig. 2.6B** and **Fig. 2.S1**). The lack of *hctB*prom, *scc2*prom, and *tarp*prom gene expression corresponded to the lack of production of infectious progeny during Pen treatment, suggesting that these genes can be considered true EB genes (**Fig. 2.6E**).

To further investigate the role of chlamydial cell division in EB development, we tested the effects of a second antibiotic that targets peptidoglycan synthesis, D-cycloserine (DCS). DCS is a cyclic analogue of D-alanine and inhibits peptidoglycan synthesis [35]. Again, *euo*prom-Clover expression was measured over time after DCS treatment at 14 hpi. The kinetics of expression of *euo*prom-Clover was similar to that of Pen-treated and untreated samples (**Fig. 2.6A** to **C**). The expression kinetics of the late gene reporters after DCS treatment also mimicked Pen treatment. DCS-treated inclusions never expressed Clover from *hctB*prom or *scc2*prom reporters but did express from the *hctA*prom reporter with a similar 9-h delay (**Fig. 2.6C**). Although the kinetics were similar to those of Pen treatment among all reporters, the aberrant RBs did not grow as large as those treated with Pen (**Fig. 2.7**).

Treatment with penicillin has been reported to induce aberrant RBs that continue to metabolize and increase in size but do not produce infectious progeny [36, 37]. Pen, other antibiotic treatments, and nutrient limitation are all reported to induce a persistent state in *Chlamydia* [38]. Therefore, we explored the effect of interferon gamma (IFN-)-induced persistence on cell-type-specific gene expression. While Pen and DCS induce persistence through their effects on peptidoglycan synthesis, IFN- causes an aberrant state by starving *Chlamydia* of tryptophan [39]. HeLa cells were used as opposed to Cos7 cells, as the former responds to human IFN- (hIFN-). Cells were treated with IFN- 24 h prior to infection with the *Ctr-ihtA*prom-EGFP or *Ctr-hctA*promClover strain. Imaging of these constructs showed that no signal was produced from either promoter construct (**Fig. 2.S2**). We also treated cells with the iron chelator bipyridyl, which is reported to have regulatory overlap of tryptophan regulation in *Chlamydia* [40]. Bipyridyl treatment also resulted in no signal produced from either promoter construct (**Fig. 2.S2**).

Data obtained from Pen- and DCS-treated infections support a role for cell division in chlamydial development. To further explore this observation, cells were treated with Pen every 2 h starting at 16 hpi. To visualize both RBs and EBs in the same inclusion during the developmental cycle, two dual promoter constructs were developed, creating *Ctr-hctA*prom-mKate2/*ihtA*prom-mNeonGreen and *Ctr-hctB*prom-mKate2/*euo*prom-Clover. Cells were infected with the dual promoter strains and imaged every 30 min starting at 14 h postinfection (**Fig. 2.8**). Expression levels of the fluorescent proteins driven by the early, early-late, and late promoters in response to Pen treatment were strikingly different. The *euo*prom signal increased compared to that of untreated infections almost immediately after Pen was added, regardless of the timing of treatment (**Fig. 2.8A**). This was also true for the other early promoter-reporter, *ihtA*prom (**Fig. 2.S3**). Signal from the late promoter *hctB*prom was completely inhibited but only after a 10-h delay, again regardless of when Pen was added (**Fig. 2.8B**). Conversely, *hctA*prom signal was inhibited very quickly after Pen treatment, but expression resumed after a 9-h delay (**Fig. 2.8C**). Confocal images of Pen-treated cells indicated that *ihtA*prom-mNeonGreen and *euo*prom-Clover expression was evident only in the large aberrant cells (**Fig. 2.9**).

However, there was a striking difference in cell type expression between the late promoters *hctA*prom and *hctB*prom. Like *ihtA*prom and *euo*prom, *hctA*prom-mKate2 expression was localized to large aberrant cells. In contrast, *hctB*prom-mKate2 expression was restricted to non-aberrant small cells that resembled EBs (**Fig. 2.9**).

**EB gene expression increases linearly until cell death.** Our data suggest that initial RB-to-EB development follows an intrinsic program and does not respond to environmental cues. However, the data show significant variability at 36 hpi. To better understand the kinetics of chlamydial development late during infection, well separated individual inclusions were monitored from when fluorescence could first be detected until lysis of the inclusion or cell. The dual promoter strain, *Ctr-hctB*promm*Kate2/euo*prom-Clover, was used to identify early inclusions and monitor late gene expression. Expression from each promoter in isolated individual inclusions was monitored for 65 hpi (**Fig. 2.10A** and **Movie 2.S2**). Late in infection, gene expression from isolated inclusions differed significantly from aggregated expression data. *euo*prom-Clover expression in each individual inclusion followed a similar pattern, a lag phase and then a short exponential phase, followed by an expression plateau at 24 hpi, which was maintained until cell death (**Fig. 2.10A** and **Movie 2.S2**). *hctB*prom expression showed a short exponential growth phase followed by continuous linear gene expression ($R^2$ 0.99) until cell lysis (**Fig. 2.10A, graph 3,** and **Movie 2.S2**). Late in infection (36 hpi), a subset of inclusions/cells lysed (**Movie 2.S2**), which contributed to the increased signal variability through loss of fluorescence, resulting in aggregate gene expression data mimicking a stationary phase. The data from single inclusions suggest that the *Chlamydia* isolates are not responding to depleting resources of the host cell late in infection, as the slope is linear until lysis. Although growth is linear for every inclusion, the rate differs between inclusions in different cells (**Fig. 2.10A, graph 2**), suggesting that the growth rate of *Chlamydia* is set by a limiting nutrient inside the cell that is maintained at a steady state, producing a linear expression curve (**Fig. 2.10A, graph 2,** and **Movie 2.S2**). Linear expression kinetics was also seen in cells grown at various temperatures. Infected cells grown at 35°C, 37°C, and 40°C all showed linear

*hctB*prom expression, with slopes varying significantly with temperature, at 344, 499, and 713 fluorescence units/h, respectively (**Fig. 2.S4**).

All data presented thus far were collected from infections in the presence of cycloheximide. Monolayers were treated with cycloheximide to block host cell division, which reduces cell migration and improves live-cell imaging. Cycloheximide is a eukaryote protein synthesis inhibitor and has been shown to increase EB production during chlamydial infections [41]. Treatment with cycloheximide is thought to decrease competition between the host and *Chlamydia* for nutrients, allowing *Chlamydia* to replicate faster [42]. To understand the impact of cycloheximide treatment on chlamydial developmental kinetics, the rates of RB and EB gene expression with and without cycloheximide were measured in individual inclusions for the entire cycle. Without cycloheximide treatment, the overall developmental pattern was retained; however, there was a delay in *euo*prom expression and a delay in the time to *euo*prom expression plateau (**Fig. 2.10B, graphs 1** and **3**). Additionally, EB gene expression in individual inclusions began later, and linear production had a significantly reduced slope (327 fluorescence units/h) in monolayers not treated with cycloheximide than in treated ones (482 fluorescence units/h) (**Fig. 2.10B, graph 3**). Interestingly, although *hctB*prom expression in the untreated cells increased at a linear rate until cell lysis, peak expression levels rarely reached that of the cycloheximide-treated cells, as cell lysis occurred before levels reached that of the treated inclusions. These data further support that EB production is a property of the growth rate and is not likely a response to changing environmental signals.

These data also suggest that growth rate of *Chlamydia* per cell is limited by steady-state levels of a limiting nutrient provided by the host, again indicating that EB development is unlikely to be linked to increasing competition or communication between *Chlamydia* but rather follows an intrinsic developmental program.

## Discussion

The infection of vertebrate hosts by *Chlamydia* is dependent on the transition between two specific cell types, the RB and EB, that each have specialized

functions. The RB undergoes cell division but is not infectious, while the EB form is responsible for mediating invasion of eukaryotic host cells and does not undergo cell division. The EB does, however, metabolize nutrients to maintain its infectious phenotype [18]. This division of labor presents a critical dilemma for *Chlamydia*, as increasing cell numbers through RB division must be balanced with the production of infectious EBs. How *Chlamydia* regulates this balance is currently unknown.

Proposed mechanisms for the control of RB-to-EB development can be divided into two broad categories, a response to extrinsic environmental cues and an intrinsic developmental program. By developing mathematical models and running simulations of infection conditions, we determined that these two possibilities could be differentiated by generating competition between RBs for environmental signals or nutrients. To explore these models experimentally, we developed a live-cell reporter system to monitor cell type switching in real time at the single-inclusion level. Cell type-specific promoters were used to drive the expression of fluorescent proteins to monitor RB growth (*ihtA*prom and *euo*prom) and EB development (*hctA*prom, *hctB*prom, *scc2*prom, and *tarp*prom). These promoter reporters were designed to detect spatial/temporal generation of fluorescence and the net of transcriptional, translational gene regulation, and maturation of the fluorophore and to not differentiate between these mechanisms. Chlamydial developmental kinetics observed using the live-cell reporter constructs were comparable to developmental data generated using qPCR for genome copies and reinfection assays to measure infectious progeny.

The use of live-cell promoter-reporters to interrogate cell type switching dramatically improved the resolution for monitoring chlamydial developmental transitions. Reporter expression was measured every 30 min at the single-inclusion level, which led to the identification of two different classes of late promoters. *hctB,* *scc2,* and *tarp* were all expressed 22 hpi and, therefore, are considered a class of true late genes. However, our data suggest that *hctA* should be considered an early-late gene, as *hctA*prom-Clover expression is induced hours before the other late genes tested and responds differently to the inhibition of chlamydial cell division. This differential timing in expression between HctA and the late proteins is

corroborated by our published RNA-seq data that demonstrated that the transcript encoding HctA was upregulated at 18 hpi, while the transcripts for HctB, Scc2, and Tarp were not detected until 24 hpi [18]. Live-cell single-inclusion analysis also highlighted the inherent limitations of endpoint population-based assays. Single inclusion dynamics demonstrated that kinetics of chlamydial development in single inclusions can be masked by cell lysis, superinfection, and reinfection in population-based studies.

Our live-cell data showed that competition for nutrients by increasing MOI and time delayed superinfections of both fusogenic and non-fusogenic inclusions, which generated competition for host cell and intra-inclusion signals and did not alter time to EB development. These data strongly suggest that development from RB to EB is independent of a competitive intra-inclusion or host environment but rather is responsive to one or more intrinsic cell-autonomous signals. Our data also showed that the developmental program is linked to a steady-state growth rate. *Chlamydia* grown at 35°C replicated slower and EB development was delayed compared to that of samples grown at 37°C. Conversely, *Chlamydia* incubated at 40°C replicated faster and EB development was initiated earlier than for growth at 37°C. Additionally, *Chlamydia* in cells treated with cycloheximide grew faster and EB development was initiated earlier than that for untreated cells.

Cell lysis and reinfection at late time points skewed the aggregate data, adding significant variability. The analysis of well-isolated single inclusions showed that each inclusion followed the same basic developmental profile. However, the *Chlamydia* in each inclusion had a unique growth rate. These data suggest that growth rate is set by steady-state kinetics in individual host cells, as EB gene (EB production) expression is linear in each cell until cell lysis but the slope varies between cells. This was also evident when comparing EB gene expression in cycloheximide-treated versus untreated host cells. The slope of *hctB*prom expression (EB production) is steeper with cycloheximide treatment, again suggesting that chlamydial growth rate is dependent on nutrient availability in the host cell. The linear kinetics of EB production suggests that *Chlamydia* does not encounter increasing nutrient limitation even toward the end of the cycle. The

kinetics of chlamydial development within individual inclusions appears to mimic that of bacteria grown in a chemostat where replication rate is controlled by a limiting nutrient. Up to a point, the host cell is actively maintaining steady-state levels of nutrients that control chlamydial growth rate and that, in turn, control EB production rate.

In addition to growth, chlamydial cell division was also required to trigger EB development. Penicillin and DCS both target peptidoglycan synthesis at different points in the pathway, resulting in a block in cell septation during chlamydial replication [43]. Both treatments, when added early in infection (prior to 14 hpi), inhibited EB formation, as measured by the production of infectious particles and expression of late gene promoter-reporters (*hctA, hctB, scc2,* and *tarp*). However, the effect of these drugs on *hctA*prom-Clover expression differed significantly from the effects seen on *hctB, scc2,* and *tarp*. Although *hctA*prom-Clover expression was initially inhibited, expression was eventually initiated in the aberrant forms after an approximately 9-h delay. We speculate this delay is the result of gene dysregulation that, over time, produces spurious regulatory outputs. Pen addition at all times tested (2-h intervals from 16 to 28 h) resulted in an immediate overall increase in *euo*prom-Clover expression and an immediate overall decrease in *hctA*prom-Clover expression in inclusions compared to untreated samples. In contrast, *hctB*prom expression kinetics was similar to that of untreated controls for approximately 10 h after Pen addition, after which point further expression was inhibited. Additionally, *hctB*prom fluorescence was only evident in small cell forms, indicating expression was restricted to EBs, while *hctA*prom expression was evident in RB-like aberrant forms, suggesting expression in an intermediate cell form. These data suggest that inhibiting cell division blocks RBs from switching off *euo*prom expression and switching on *hctA*prom gene expression. However, if a cell is already committed to EB formation (*hctA*prom positive), EB gene expression continues (Pen insensitive) until the EB is fully mature (maximal *hctB*prom signal), which our data indicate takes about 10 h in *C. trachomatis* L2.

The treatment of *Chlamydia*-infected cells with penicillin, other antibiotics, or reagents that cause nutrient limitation results in a growth phenotype termed

persistence [38]. Persistence is characterized by aberrant RB forms that are larger than untreated RBs, do not undergo cell division, and do not produce infectious progeny [38]. Although all these treatments cause aberrant RBs, the phenotypes vary [39, 44]. Pen and DCS treatment cause persistence by inhibiting cell division through inhibiting peptidoglycan synthesis, while IFN- treatment causes persistence by inducing the enzyme indoleamine-2,3-dioxygenase in the host cell, which serves to deplete tryptophan levels in the cell, starving *Chlamydia* of this essential amino acid[39]. Comparing the live-cell imaging data from these different persistence inducers revealed that the IFN--treated *Chlamydia* never expressed Clover from any promoters tested early or late. This was also true for *Chlamydia* grown in the presence of the iron chelator bipyridyl. The *Chlamydia* from bipyridyl-treated infections never expressed the fluorescent reporters from early or late promoters. This dramatic difference in gene regulation suggests different mechanisms are involved and that persistence is not a phenotype associated with a specific gene expression profile.

Overall, our data support a model in which RB-to-EB development follows a cell-autonomous preprogrammed cycle that requires chlamydial division. Our initial mathematical models assumed an inhibitory signal that, at high concentrations, inhibited RBs from differentiating into EBs. The concentration of this signal was depleted by metabolic utilization, and RB-to-EB differentiation occurred. We have now updated this model to reflect our current data supporting an intrinsic signal linked to chlamydial growth rate and cell division. This model suggests the involvement of an internal signal in the nascent RB that, at high concentrations, inhibits RBs from differentiating into EBs, and that the signal concentration is depleted through dilution via 3 to 5 cell divisions and not metabolic utilization. After the inhibitory signal is reduced below a threshold, RBs are capable of transitioning to EBs (**Fig. 2.11**). Of the current proposed models in the literature (nutrient limitation [45], inclusion membrane limitation [46], and RB size [14]), only the model based on RB size is consistent with our data. The RB size model described by Lee et al. proposed that RB growth rate is lower than the division rate, leading to a size reduction (depletion of signal) of the RBs after each division. After several rounds of division, a

size threshold is reached and EB development is triggered [14]. This proposed mechanism fits our model, as size would act as the inhibitory signal that is reduced through cell division. It should be noted that although we propose the dilution of an inhibitor as the intrinsic signal to control cell type switching, it is equally possible that a positive signal linked to cell division, such as the development of asymmetry/ polarity, could act as an EB-promoting signal.

Chlamydial development can be considered to occur in two steps, an RB exponential growth step starting 12 hpi (*C. trachomatis* serovar L2) and an asynchronous EB production step starting at 18 hpi (*C. trachomatis* serovar L2) [47, 48]. Although the size reduction model and our model explain some of the gene expression patterns that control cell type switching, it is clear that EB development is more complicated than these simple switch models. The output of the models fit the switch between the RB exponential growth phase and the beginning of EB development, but they do not adequately explain the continued requirement for cell division during asynchronous EB production. Our data show that Pen treatment blocks the *euo*-to-*hctA* gene expression switch even when added late in infection (28 hpi), well after the time of initial EB formation (18 hpi). Further evidence for a dilution-independent second step is the observation that the *euo*prom-to-*hctA*prom switch is initially blocked by both DCS treatment and Pen treatment, yet this inhibition is eventually overcome and *hctA*prom-Clover is expressed after a 9-h delay. Unlike Pen treatment, where RBs continue to increase in size, DCS impacts cell growth, resulting in smaller RBs and, thus, limiting the effect of dilution. These observations support a second developmental regulatory step that is independent of inhibitor dilution, suggesting cell division itself is an important step in committing to the EB cell type.

Our interpretation of these data is that EB formation is multifactorial and requires multiple steps to form a final infectious EB. The first step is the loss of the inhibitory signal in the RB through multiple rounds of division, where early RBs (RBR) divide 3 to 5 times by binary fission, eventually becoming competent to produce EBs (RBE). This is followed by a second step that is dependent on asymmetric cell division creating two cells with different expression profiles. One daughter cell remains an

RBE (*euo*prom positive), and the second daughter cell becomes committed to EB formation (IB, *hctA*prom positive) (**Fig. 2.11**). The committed IB cell (*hctA*prom positive) does not divide but matures into the infectious EB (*hctB*prom, *scc2*prom, and *tarp*prom positive). Further divisions of the RBE cell produce one RBE and one IB leading to the linear increase in EBs that we report. The data from the Pen treatment experiments also suggest that EB maturation, from *hctA*prom positive to *hctB*prom positive, takes 8 to 10 h, but we do not yet know when, along this progression, infectivity is gained.

Additional support for asymmetric EB production is the observation that *hctB*prom signal (EB production) follows a nearly perfect linear trajectory and is not logarithmic during the EB production phase (24 hpi; cell lysis) (**Fig. 2.10A, Fig. 2.S4, and Movie 2.S2**). In contrast, the *euo*prom signal (RB growth) transitions from log to linear to no growth (**Fig. 2.10A** and **Movie 2.S2**). These observations suggest that the RBR cell population expands by exponential growth followed by a transition to the RBE cell type. The RBE then divides asymmetrically, leading to EB production with no gain in RBE numbers. Asymmetric cell division producing two cells with differing fates is reminiscent of stalk/swarmer cell systems best described in *Caulobacter crescentus* [49] but also described in the *Planctomycetes* genus that is more closely related to *Chlamydia* [50]. This is also supported by other studies that have provided evidence for asymmetric cell division in *C. trachomatis*. These studies show that the cell division machinery assembles asymmetrically, leading to polarized RB division [43, 51, 52]. Additionally, the EB itself is asymmetric, demonstrating hemispherical projections that can be seen by electron microscopy [53].

Overall, our data show that the combination of mathematical modeling and live-cell gene reporter imaging is a powerful tool to tease apart the molecular details of cell type development. Continued revision and testing of our models of development will lead to an expanded understanding of cell type development in this important human pathogen.

## Materials and Methods

**Organisms and cell culture.** Cos-7 and HeLa cells were obtained from the American Type Culture Collection (ATCC). Cos-7 cells were used for all experiments unless otherwise specified. Both Cos-7 and HeLa cells were maintained in a 5% $CO_2$ incubator at 37°C (unless otherwise indicated) in RPMI 1640 (Cellgro) supplemented with 10% fetal plex and 10 g/ml gentamicin. All *C. trachomatis* L2 (LGV 434) strains were grown in and harvested from Cos-7 cells. Elementary bodies were purified by density centrifugation using 30% MD-76R 48 h post infection [18]. Purified elementary bodies were stored at 80°C in sucrose-phosphate-glutamate buffer (10 mM sodium phosphate [8 mM K2HPO4, 2 mM KH2PO4], 220 mM sucrose, 0.50 mM L-glutamic acid; pH 7.4). Escherichia coli ER2925 (mutated in dam and dcm) was utilized to produce unmethylated constructs for transformation into *Chlamydia*.

**Reporter plasmids.** The backbone for all promoter-reporter constructs was p2TK2SW2 [54]. Promoters were amplified from *C. trachomatis* L2 genomic DNA using the primers indicated (see Table 3.1 in the supplemental material). Each promoter sequence consisted of 100 bp upstream of the predicted transcription start site for the specified chlamydial genes plus the untranslated region and the first 30 nt (10 amino acids) of the respective ORF. Promoter sequences were inserted into p2TK2SW2 downstream of the ColE1 ORI. Fluorescent reporters (EGFP/Clover/mNeonGreen/mKate2) were ordered as gene blocks from Integrated DNA Technologies (IDT) and inserted in frame with the first 30 nt of the chlamydial gene. Each ORF was followed by the incD terminator. The bla gene was replaced by the aadA gene (spectinomycin resistance) from pBam4. The final constructs reported in this study were p2TK2-*ihtA*prom-EGFP, p2TK2-*hctA*prom-Clover, p2TK2-*hctB*prom-Clover, p2TK2- *scc*2prom-Clover, p2TK2-*euo*prom-Clover, p2TK2-*tarp*prom-Clover, p2TK2-*hctB*prom-mKate2/*euo*promClover, and p2TK2-*hctA*prom-mKate2/*ihtA*prom-mNeonGreen.

**Chlamydial transformation and isolation.** Transformation of *C. trachomatis* L2 was performed as previously described [54] and selected using 500 ng/l spectinomycin. Clonal isolation was achieved via successive rounds of inclusion

isolation (MOI, 1) using a micromanipulator. The plasmid constructs were purified from chlamydial transformants, transformed into *E. coli*, and sequenced.

**Infections.** To synchronize infections, host cells were incubated with *C. trachomatis* EBs in Hanks' balanced salt solution (HBSS) (Gibco) for 15 min at 37°C with rocking. The inoculum was removed and cells were washed with prewarmed (37°C) 1 mg/ml heparin sodium in HBSS. The HBSS with heparin was replaced with fresh RPMI 1640 containing 10% fetal bovine serum, 10 g/ml gentamicin, and 1 g/ml cycloheximide, unless otherwise stated. For cell division experiments, chlamydial cell division was inhibited by the addition of 1 U/ml penicillin G or 40 g/ml D-cycloserine to the media. To starve *Chlamydia* of tryptophan, HeLa cells were incubated for 24 h in medium containing 2 ng/ml recombinant human IFN-(PHC4033; Invitrogen) prior to infection. Iron starvation of *Chlamydia* was achieved by treating Cos-7 cells with the iron chelator bipyridyl (100 M) upon infection with *Ctr*-L2-prom EBs [55].

**Replating assays.** *Ctr-hctA*prom-Clover EBs were obtained from infected Cos-7 cells by scraping the host monolayer and pelleting via centrifugation for 30 min at 17,200 relative centrifugal force. The EB pellets were resuspended in RPMI via sonication. For reinfection, Cos-7 cells were plated to confluence in clear polystyrene 96-well microplates. EB reinfections consisted of 2-fold dilutions. Spectinomycin was added to superinfection experiments to prevent wild-type *C. trachomatis* L2 growth. Infected plates were incubated for 29 h. Cells were fixed with methanol and stained with 4,6-diamidino-2-phenylindole (DAPI). The DAPI stain was used for automated microscope focus and visualization of host cell nuclei, and GFP-Clover was used for visualization of EBs and inclusion counts. Inclusions were imaged using a Nikon Eclipse TE300 inverted microscope utilizing a scopeLED lamp at 470 nm and 390 nm and BrightLine bandpass emissions filters at 514/30 nm and 434/17 nm. Image acquisition was performed using an Andor Zyla sCMOS in conjunction with Manager software. Images were analyzed using ImageJ software [56] and custom scripts (**Supplemental Material 2.S2**).

**Genome number quantification.** Chlamydial genomic DNA was isolated from infected host cells during active infections using an Invitrogen PureLink genomic

DNA mini kit. An ABI-7900HT reverse transcription PCR system was utilized for the quantification of genomic copies. A DyNAmo Flash SYBR green qPCR kit and *hctA*-specific primer were used for detection.

**Fluorescence microscopy.** Cos-7 monolayers were infected with synchronized *Ctr*-L2-prom EBs. Live infections were grown in an OKOtouch $CO_2$/heated stage incubator. Infections were imaged using a Nikon Eclipse TE300 inverted microscope using epifluorescence imaging and a 20, 0.4-numericaperture objective, giving a depth of field of about 5.8 m. A ScopeLED lamp at 470 nm and 595 nm and BrightLine bandpass filters at 514/30 nm and 590/20 nm were used for excitation and emission. DIC was used for focus. Image acquisition was performed using an Andor Zyla sCMOS camera in conjunction with Manager software [57]. Images were taken at 30-min intervals from 10 to 48 h after *Ctr*-L2-prom infection unless otherwise stated. Live-cell infections were performed in 24- or 96-well glass-bottom plates, allowing treatments to vary between wells. Multiple fields were imaged for each treatment. Fluorescent intensities for individual inclusions were monitored over time using the Trackmate plug-in in ImageJ[22]. Inclusion fluorescent intensities were then analyzed and graphed using pandas, matplotlib, and seaborn in custom Python notebooks. The scripts for this analysis are available from the github account (https://github.com/SGrasshopper).

For confocal microscopy, samples were fixed with 4% paraformaldehyde, washed with phosphate buffered saline, and mounted with MOWIOL. Confocal images were acquired using a Nikon spinning disk confocal system with a 60 oil immersion objective, equipped with an Andor Ixon electron-multiplying charge-coupled device camera under the control of Nikon Elements software. Images were processed using the image analysis software ImageJ (http://rsb.info.nih.gov/ij/). Representative confocal micrographs displayed in the figures are maximal intensity projections of the three-dimensional data sets unless otherwise noted.

**Data availability.** All data, bacterial strains, and methodologies are available upon request.

**References**

1.  Borel N, Polkinghorne A, Pospischil A. 2018. A review on chlamydial diseases in animals: still a challenge for pathologists? Vet Pathol 55: 374 –390. https://doi.org/10.1177/0300985817751218.

2.  Zhang J, Lietman T, Olinger L, Miao Y, Stephens RS. 2004. Genetic diversity of *Chlamydia trachomatis* and the prevalence of trachoma. Pediatr Infect Dis J 23:217–220. https://doi.org/10.1097/01.inf.0000115501.60397.a6.

3.  Owusu-Edusei KJ, Chesson HW, Gift TL, Tao G, Mahajan R, Ocfemia MCB, Kent CK. 2013. The estimated direct medical cost of selected sexually transmitted infections in the United States, 2008. Sex Transm Dis 40: 197–201. https://doi.org/10.1097/OLQ.0b013e318285c6d2.

4.  Owusu-Edusei K, Roby TM, Chesson HW, Gift TL. 2013. Productivity costs of nonviral sexually transmissible infections among patients who miss work to seek medical care: evidence from claims data. Sex Health 10:434 – 437. https://doi.org/10.1071/SH13021.

5.  Torrone EA, Bernstein KT. 2014. Surveillance for sexually transmitted diseases, p 122–131. In Concepts and methods in infectious disease surveillance. John Wiley & Sons, Ltd, Chichester, United Kingdom.

6.  Datta SD, Torrone E, Kruszon-Moran D, Berman S, Johnson R, Satterwhite CL, Papp J, Weinstock H. 2012. *Chlamydia trachomatis* trends in the United States among persons 14 to 39 years of age, 1999 –2008. Sex Transm Dis 39:92–96. https://doi.org/10.1097/OLQ.0b013e 31823e2ff7.

7.  Ohman H, Tiitinen A, Halttunen M, Lehtinen M, Paavonen J, Surcel H-M. 2009. Cytokine polymorphisms and severity of tubal damage in women with *Chlamydia*-associated infertility. J Infect Dis 199:1353–1359. https:// doi.org/ 10.1086/597620.

8.  Bébéar C, de Barbeyrac B. 2009. Genital *Chlamydia trachomatis* infections. Clin Microbiol Infect 15:4 –10. https://doi.org/10.1111/j.1469-0691 .2008.02647.x.

9.  Miller WC, Ford CA, Morris M, Handcock MS, Schmitz JL, Hobbs MM, Cohen MS, Harris KM, Udry JR. 2004. Prevalence of chlamydial and gonococcal

infections among young adults in the United States. JAMA 291:2229 –2236. https://doi.org/10.1001/jama.291.18.2229.

10. World Health Organization. 2001. Global prevalence and incidence of selected curable sexually transmitted infections: overview and estimates. World Health Organization, Geneva, Switzerland.

11. Abdelrahman YM, Belland RJ. 2005. The chlamydial developmental cycle. FEMS Microbiol Rev 29:949 –959. https://doi.org/10.1016/j.femsre.2005 .03.002.

12. Clifton DR, Fields KA, Grieshaber SS, Dooley CA, Fischer ER, Mead DJ, Carabeo RA, Hackstadt T. 2004. A chlamydial type III translocated protein is tyrosine-phosphorylated at the site of entry and associated with recruitment of actin. Proc Natl Acad SciUSA 101:10166 –10171. https:// doi.org/10.1073/ pnas.0402829101.

13. Scidmore MA, Rockey DD, Fischer ER, Heinzen RA, Hackstadt T. 1996. Vesicular interactions of the *Chlamydia trachomatis* inclusion are determined by chlamydial early protein synthesis rather than route of entry. Infect Immun 64:5366 –5372. https://doi.org/10.1128/IAI.64.12 .5366-5372.1996.

14. Lee JK, Enciso GA, Boassa D, Chander CN, Lou TH, Pairawan SS, Guo MC, Wan FYM, Ellisman MH, Sütterlin C, Tan M. 2018. Replication-dependent size reduction precedes differentiation in *Chlamydia trachomatis*. Nat Commun 9:45. https://doi.org/10.1038/s41467-017-02432-0.

15. Hackstadt T, Fischer ER, Scidmore MA, Rockey DD, Heinzen RA. 1997. Origins and functions of the chlamydial inclusion. Trends Microbiol 5:288 –293. https:// doi.org/10.1016/S0966-842X(97)01061-5.

16. Hoare A, Timms P, Bavoil PM, Wilson DP. 2008. Spatial constraints within the chlamydial host cell inclusion predict interrupted development and persistence. BMC Microbiol 8:5. https://doi.org/10.1186/1471-2180-8-5.

17. Omsland A, Sager J, Nair V, Sturdevant DE, Hackstadt T. 2012. Developmental stage-specific metabolic and transcriptional activity of *Chlamydia trachomatis* in an axenic medium. Proc Natl Acad SciUSA 109:19781–19785. https://doi.org/ 10.1073/pnas.1212831109.

18. Grieshaber S, Grieshaber N, Yang H, Baxter B, Hackstadt T, Omsland A. 2018. Impact of active metabolism on *Chlamydia trachomatis* elementary body transcript profile and infectivity. J Bacteriol 200:563. https:// doi.org/10.1128/ JB.00065-18.

19. Grieshaber NA, Grieshaber SS, Fischer ER, Hackstadt T. 2006. A small RNA inhibits translation of the histone-like protein Hc1 in *Chlamydia trachomatis*. Mol Microbiol 59:541–550. https://doi.org/10.1111/j.1365-2958 .2005.04949.x.

20. Barry CE, Brickman TJ, Hackstadt T. 1993. Hc1-mediated effects on DNA structure: a potential regulator of chlamydial development. Mol Microbiol 9:273–283. https://doi.org/10.1111/j.1365-2958.1993.tb01689.x.

21. Grieshaber NA, Tattersall JS, Liguori J, Lipat JN, Runac J, Grieshaber SS. 2015. Identification of the base-pairing requirements for repression of hctA translation by the small RNA IhtA leads to the discovery of a new mRNA target in *Chlamydia trachomatis*. PLoS One 10:e0116593. https:// doi.org/10.1371/ journal.pone.0116593.

22. Tinevez J-Y, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, Bednarek SY, Shorte SL, Eliceiri KW. 2017. TrackMate: an open and extensible platform for single-particle tracking. Methods 115:80 –90. https://doi.org/10.1016/ j.ymeth.2016.09.016.

23. Chiarelli TJ, Grieshaber NA, Grieshaber SS. 2020. Live-cell forward genetic approach to identify and isolate developmental mutants in *Chlamydia trachomatis*. J Vis Exp 160:e61365. https://doi.org/10.3791/61365.

24. Ratkowsky DA, Olley J, McMeekin TA, Ball A. 1982. Relationship between temperature and growth rate of bacterial cultures. J Bacteriol 149:1–5. https:// doi.org/10.1128/JB.149.1.1-5.1982.

25. Joubert BC, Sturm AW. 2011. Differences in *Chlamydia trachomatis* growth rates in human keratinocytes among lymphogranuloma venereum reference strains and clinical isolates. J Med Microbiol 60: 1565–1569. https://doi.org/10.1099/ jmm.0.032169-0.

26. Hackstadt T, Brickman TJ, Barry CE, Sager J. 1993. Diversity in the *Chlamydia trachomatis* histone homologue Hc2. Gene 132:137–141. https://doi.org/10.1016/0378-1119(93)90526-9.

27. Fields KA, Fischer ER, Mead DJ, Hackstadt T. 2005. Analysis of putative *Chlamydia trachomatis* chaperones Scc2 and Scc3 and their use in the identification of type III secretion substrates. J Bacteriol 187:6466 – 6478. https://doi.org/10.1128/JB.187.18.6466-6478.2005.

28. Grieshaber SS, Grieshaber NA, Hackstadt T. 2003. *Chlamydia trachomatis* uses host cell dynein to traffic to the microtubule-organizing center in a p50 dynamitin-independent process. J Cell Sci 116:3793–3802. https:// doi.org/10.1242/jcs.00695.

29. Hackstadt T, Scidmore-Carlson MA, Shaw EI, Fischer ER. 1999. The *Chlamydia trachomatis* IncA protein is required for homotypic vesicle fusion. Cell Microbiol 1:119 –130. https://doi.org/10.1046/j.1462-5822.1999.00012.x.

30. Richards TS, Knowlton AE, Grieshaber SS. 2013. *Chlamydia trachomatis* homotypic inclusion fusion is promoted by host microtubule trafficking. BMC Microbiol 13:185. https://doi.org/10.1186/1471-2180-13-185.

31. Suchland RJ, Rockey DD, Bannantine JP, Stamm WE. 2000. Isolates of *Chlamydia trachomatis* that occupy non-fusogenic inclusions lack IncA, a protein localized to the inclusion membrane. Infect Immun 68:360 –367. https://doi.org/10.1128/iai.68.1.360-367.2000.

32. Liechti G, Kuru E, Packiam M, Hsu Y-P, Tekkam S, Hall E, Rittichier JT, VanNieuwenhze M, Brun YV, Maurelli AT. 2016. Pathogenic chlamydia lack a classical sacculus but synthesize a narrow, mid-cell peptidoglycan ring, regulated by MreB, for cell division. PLoS Pathog 12:e1005590. https://doi.org/10.1371/journal.ppat.1005590.

33. Rosario CJ, Tan M. 2012. The early gene product EUO is a transcriptional repressor that selectively regulates promoters of *Chlamydia* late genes. Mol Microbiol 84:1097–1107. https://doi.org/10.1111/j.1365-2958.2012 .08077.x.

34. Lambden PR, Pickett MA, Clarke IN. 2006. The effect of penicillin on *Chlamydia trachomatis* DNA replication. Microbiology 152:2573–2578. https://doi.org/10.1099/mic.0.29032-0.

35. Prosser GA, de Carvalho LPS. 2013. Kinetic mechanism and inhibition of Mycobacterium tuberculosis D-alanine: D-alanine ligase by the antibiotic D-cycloserine. FEBS J 280:1150 –1166. https://doi.org/10 .1111/febs.12108.

36. Bavoil PM. 2012. What's in a word: the use, misuse, and abuse of the word "persistence" in *Chlamydia* biology. Front Cell Infect Microbiol 4:27. https://doi.org/10.3389/fcimb.2014.00027.

37. Matsumoto A, Manire GP. 1970. Electron microscopic observations on the effects of penicillin on the morphology of *Chlamydia psittaci*. J Bacteriol 101:278 –285. https://doi.org/10.1128/JB.101.1.278-285 .1970.

38. Panzetta ME, Valdivia RH, Saka HA. 2018. *Chlamydia* persistence: a survival strategy to evade antimicrobial effects in vitro and in vivo. Front Microbiol 9:3101. https://doi.org/10.3389/fmicb.2018.03101.

39. Beatty WL, Byrne GI, Morrison RP. 1993. Morphologic and antigenic characterization of interferon gamma-mediated persistent *Chlamydia trachomatis* infection in vitro. Proc Natl Acad SciUSA 90:3998 – 4002. https://doi.org/10.1073/pnas.90.9.3998.

40. Pokorzynski ND, Brinkworth AJ, Carabeo R. 2019. A bipartite iron dependent transcriptional regulation of the tryptophan salvage pathway in *Chlamydia trachomatis*. Elife 8:e42295. https://doi.org/10.7554/ eLife.42295.

41. Ripa KT, Mårdh PA. 1977. Cultivation of *Chlamydia trachomatis* in cycloheximide-treated mccoy cells. J Clin Microbiol 6:328 –331.

42. Sabet SF, Simmons J, Caldwell HD. 1984. Enhancement of *Chlamydia trachomatis* infectious progeny by cultivation of HeLa 229 cells treated with DEAE-dextran and cycloheximide. J Clin Microbiol 20:217–222. https://doi.org/10.1128/JCM.20.2.217-222.1984.

43. Liechti GW, Kuru E, Hall E, Kalinda A, Brun YV, VanNieuwenhze M, Maurelli AT. 2014. A new metabolic cell-wall labelling method reveals peptidoglycan in

*Chlamydia trachomatis*. Nature 506:507–510. https:// doi.org/10.1038/ nature12892.

44. Weiss E. 1950. The effect of antibiotics on agents of the psittacosis lymphogranuloma group. I. The effect of penicillin. J Infect Dis 87: 249 –263. https://doi.org/10.1093/infdis/87.3.249.

45. Omsland A, Sixt BS, Horn M, Hackstadt T. Chlamydial metabolism revisited: interspecies metabolic variability and developmental stage specific physiologic activities. FEMS Microbiol Rev 38:779 – 801. https:// doi.org/ 10.1111/1574-6976.12059.

46. Wilson DP, Timms P, McElwain DLS, Bavoil PM. 2006. Type III secretion, contact-dependent model for the intracellular development of *Chlamydia*. Bull Math Biol 68:161–178. https://doi.org/10.1007/s11538-005 -9024-1.

47. Wyrick PB. 2010. *Chlamydia trachomatis* persistence in vitro: an overview. J Infect Dis 201:88 –95. https://doi.org/10.1086/652394.

48. Shaw EI, Dooley CA, Fischer ER, Scidmore MA, Fields KA, Hackstadt T. 2000. Three temporal classes of gene expression during the *Chlamydia trachomatis* developmental cycle. Mol Microbiol 37:913–925. https://doi .org/10.1046/ j.1365-2958.2000.02057.x.

49. Collier J. 2019. Cell division control in Caulobacter crescentus. Biochim Biophys Acta Gene Regul Mech 1862:685– 690. https://doi.org/10.1016/ j.bbagrm.2018.04.005.

50. Lee K-C, Webb RI, Fuerst JA. 2009. The cell cycle of the planctomycete Gemmata obscuriglobus with respect to cell compartmentalization. BMC Cell Biol 10:4. https://doi.org/10.1186/1471-2121-10-4.

51. Abdelrahman Y, Ouellette SP, Belland RJ, Cox JV. 2016. Polarized cell division of *Chlamydia trachomatis*. PLoS Pathog 12:e1005822. https:// doi.org/10.1371/ journal.ppat.1005822.

52. Ouellette SP, Lee J, Cox JV. 2020. Division without binary fission: cell division in the FtsZ-less *Chlamydia*. J Bacteriol 202:e00252-20. https:// doi.org/10.1128/ JB.00252-20.

53. Gregory WW, Gardner M, Byrne GI, Moulder JW. 1979. Arrays of hemispheric surface projections on *Chlamydia psittaci* and *Chlamydia trachomatis* observed by scanning electron microscopy. J Bacteriol 138: 241–244. https://doi.org/10.1128/JB.138.1.241-244.1979.

54. Wang Y, Kahane S, Cutcliffe LT, Skilton RJ, Lambden PR, Clarke IN. 2011. Development of a transformation system for *Chlamydia trachomatis*: restoration of glycogen biosynthesis by acquisition of a plasmid shuttle vector 9. PLoS Pathog 7:e1002258. https://doi.org/10.1371/journal.ppat.1002258.

55. Thompson CC, Carabeo RA. 2009. An optimal method of iron starvation of the obligate intracellular pathogen, *Chlamydia trachomatis*. Front Microbiol 2:20. https://doi.org/10.3389/fmicb.2011.00020.

56. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez J-Y, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. 2012. Fiji: an open source platform for biological-image analysis. Nat Methods 9:676 – 682. https://doi.org/10.1038/nmeth.2019.

57. Edelstein AD, Tsuchida MA, Amodaj N, Pinkard H, Vale RD, Stuurman N. 2014. Advanced methods of microscope control using Manager software. J Biol Methods 1:10. https://doi.org/10.14440/jbm.2014.36.
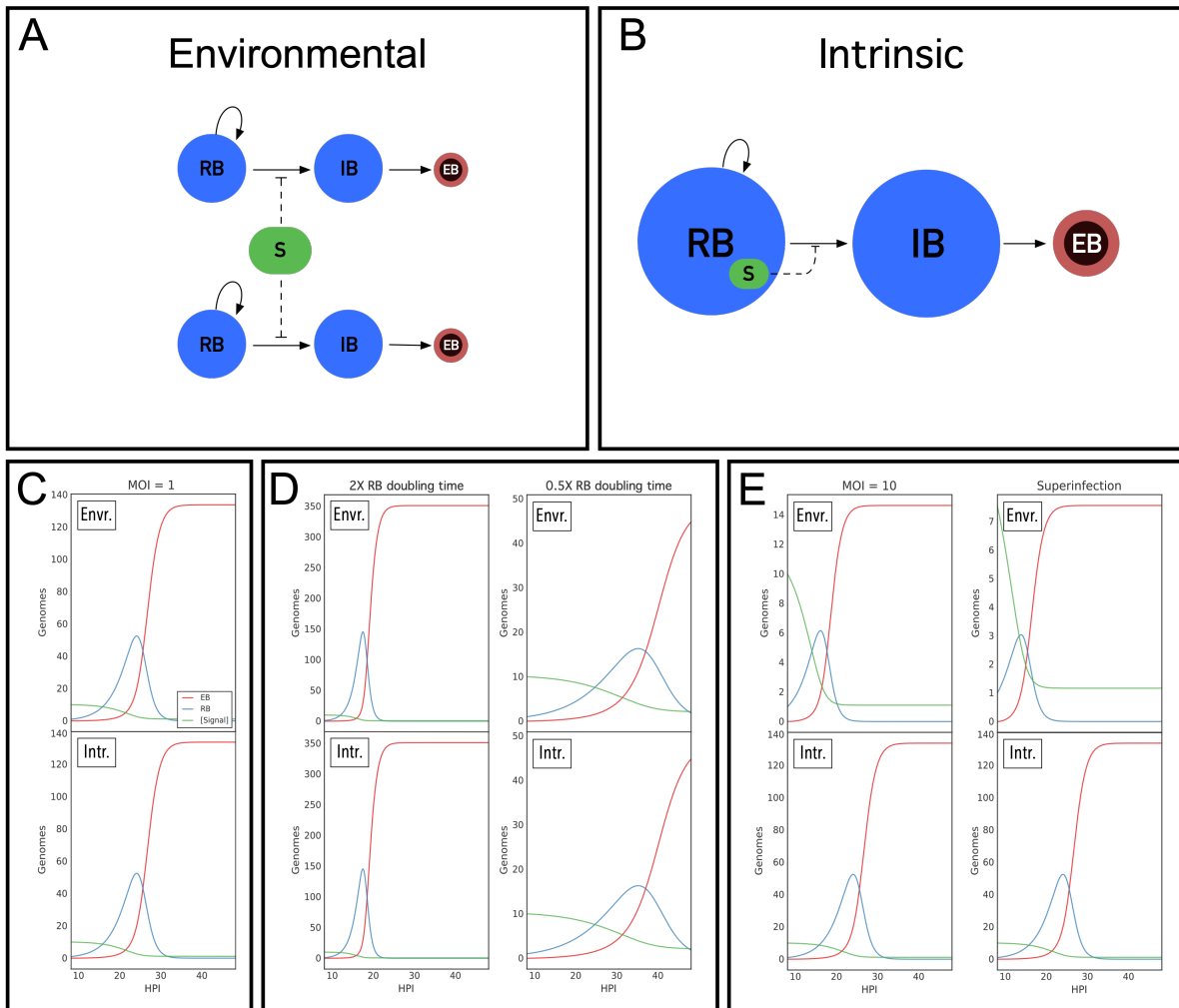
**Figure 2.1**

**Figure 2.1: Schematic and simulations of environmental and intrinsic models.**

Both models assume that the mechanism of RB/EB conversion is in response to signal concentration. High signal concentration prevents RB/EB conversion, and RB replication continues. As RBs replicate, the signal is consumed. Once the signal is depleted past a given threshold, RBs convert to IBs, which then convert to EBs. (A) Schematic of the environmental signal model. The RBs compete for a single pool of signal (S). (B) Schematic of the intrinsic model. Each RB contains its own signal, eliminating competition between RBs. (C) Simulations of the two models (environmental and intrinsic) using a multiplicity of infection (MOI) of 1 and an RB generation time of 2.27 h produced results that mimic the general kinetics of the chlamydial cycle and were indistinguishable from each other. (D) Simulations of RB doubling times of 1.13 h (half the measured RB doubling time) resulted in a reduced time to EB production, whereas 4.54 h (2 the measured RB doubling time) increased time to EB production. However, both models (environmental and intrinsic) produced the same outcome. (E) Simulations using an MOI of 10 predicted EB conversion to occur more rapidly in the environmental signal model but to remain unchanged in the intrinsic model. Similarly, simulations of the models using a time-delayed superinfection resulted in RB-to-EB conversion occurring more rapidly in the environmental model but remaining unchanged in the intrinsic model.

**Figure 2.2**
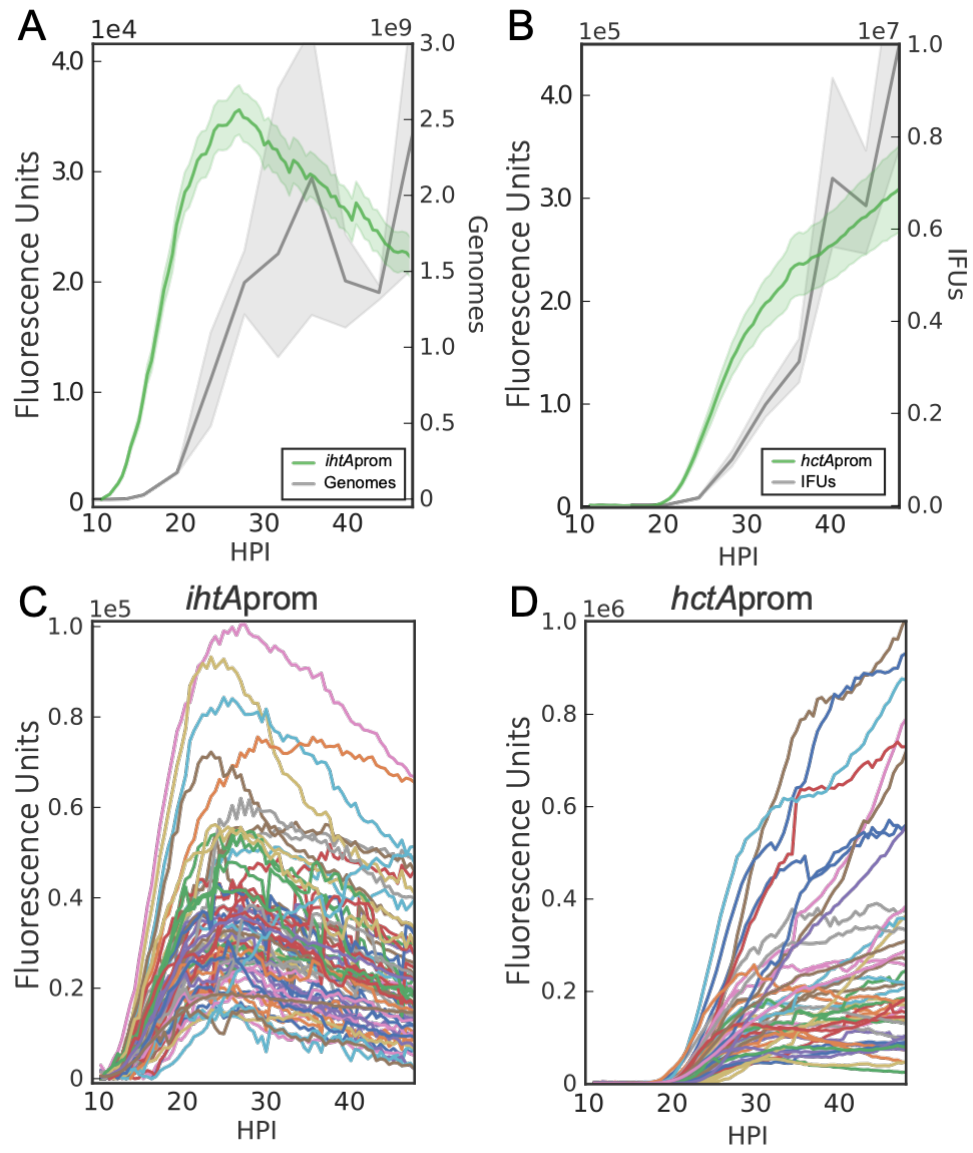
**Figure 2.2: Live-cell fluorescent imaging of chlamydial development.**

Cell type-specific fluorescent reporters were created to track chlamydial development in real time. Infections with purified *Ctr*-L2-prom EBs were synchronized and fluorescence microscopy, and qPCR/reinfection assays were run simultaneously. (A and B) The averages of *ihtA*prom-EGFP and *hctA*prom-Clover expression intensities from 50 individual inclusions monitored via automated live-cell fluorescence microscopy throughout the developmental cycle compared to genome copies and IFU, respectively. (C and D) The fluorescence intensities of 50 individual inclusions tracked via live-cell microscopy throughout the developmental cycle. The fluorescent unit cloud represents standard error of the mean (SEM) genome copies, and the IFU cloud represents 95% confidence intervals (CI). y axes are denoted in scientific notation.

**Figure 2.3**
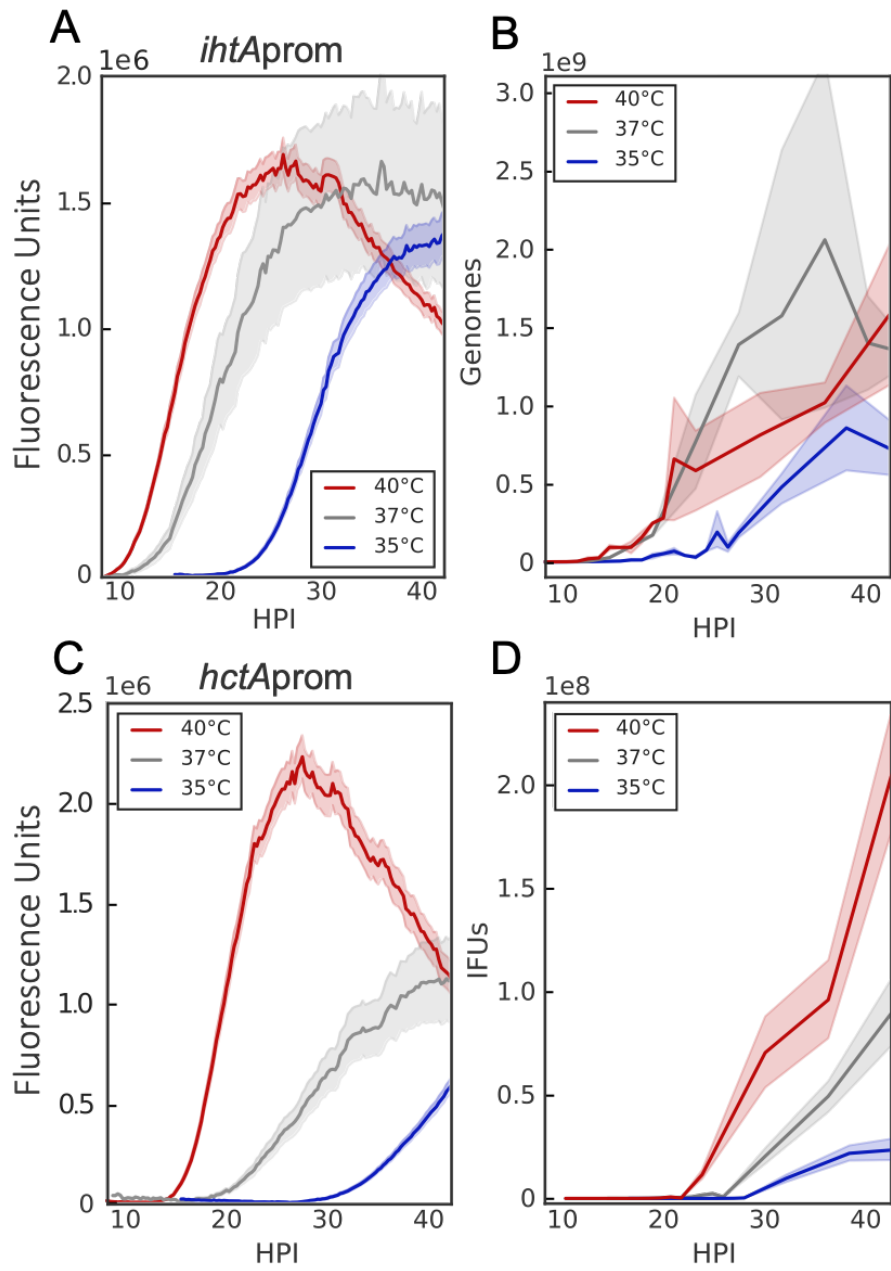
**Figure 2.3: RB replication and EB conversion are growth rate dependent.**

The ability of the promoter-reporter system to monitor differences in RB replication and EB conversion was tested by altering the growth temperature (35°C, blue; 37°C, gray; 40°C, red). (A) The averages of *ihtA*prom-EGFP expression intensities of 50 individual inclusions monitored from 9 to 42 hpi via live-cell fluorescence microscopy. (B) Genome copies were measured between 2 and 42 hpi by qPCR. (C) The averages of *hctA*prom-Clover expression intensities of 50 individual inclusions monitored from 9 to 42 hpi via live-cell fluorescence microscopy. (D) EB conversion (IFU) was quantified via replating assay from 11 to 42 hpi. The fluorescent unit cloud represents standard error of the mean (SEM) genome copies, and the IFU cloud represents 95% confidence intervals (CI). y axes are denoted in scientific notation.

**Figure 2.4**
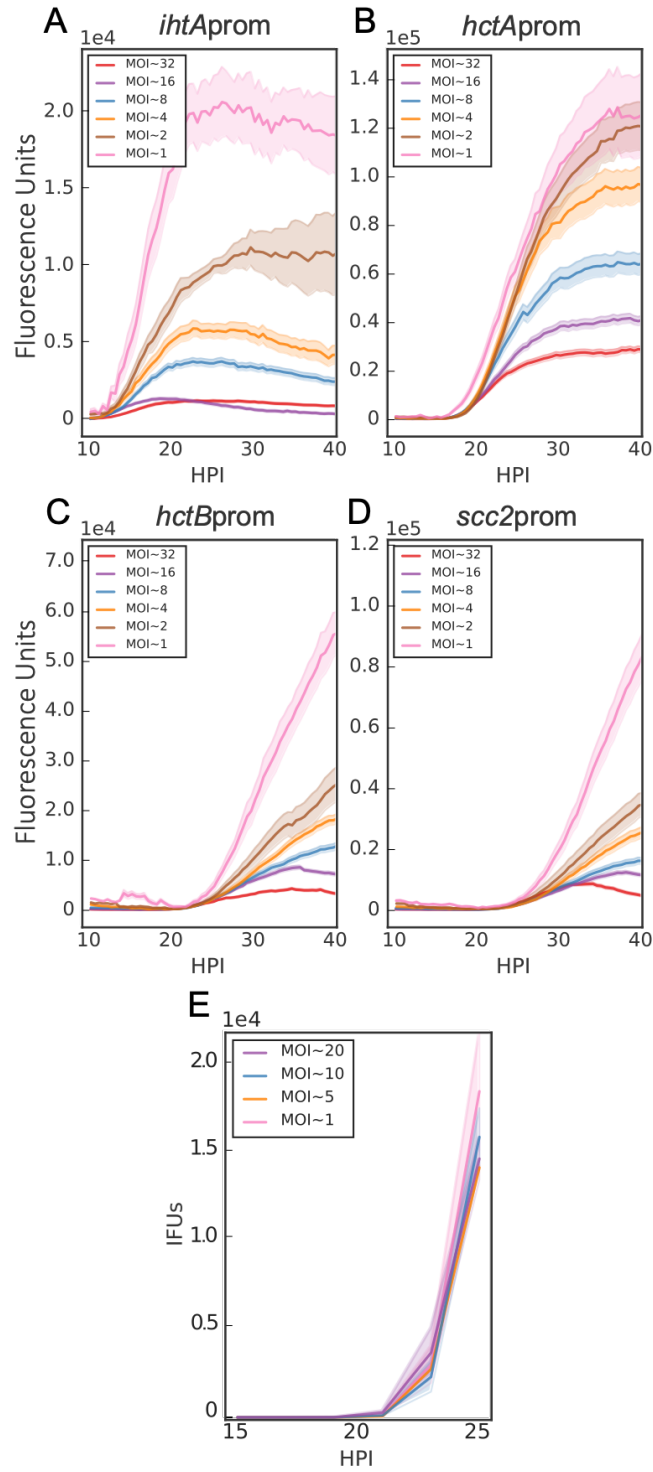
**Figure 2.4: MOI does not affect initiation of RB-to-EB conversion.**

Host cells were infected with purified *Ctr*-L2-prom EBs at an MOI of 1 to 32. (A to D) Averages of *ihtA*prom-EGFP, *hctA*prom-Clover, *hctB*prom-Clover, and *scc2*prom-Clover expression intensities from 50 individual inclusions monitored via automated live-cell fluorescence microscopy throughout the developmental cycle. Fluorescent intensities were normalized by the respective MOI. (E) EB development (IFU) was measured at MOIs from 1 to 20 and was quantified via a replating assay. EBs were harvested at 2-h intervals from 15 to 25 hpi. IFU data were normalized by the respective MOI. The fluorescent unit cloud represents standard error of the mean (SEM) genome copies, and the IFU cloud represents 95% confidence intervals (CI). y axes are denoted in scientific notation.

**Figure 2.5**
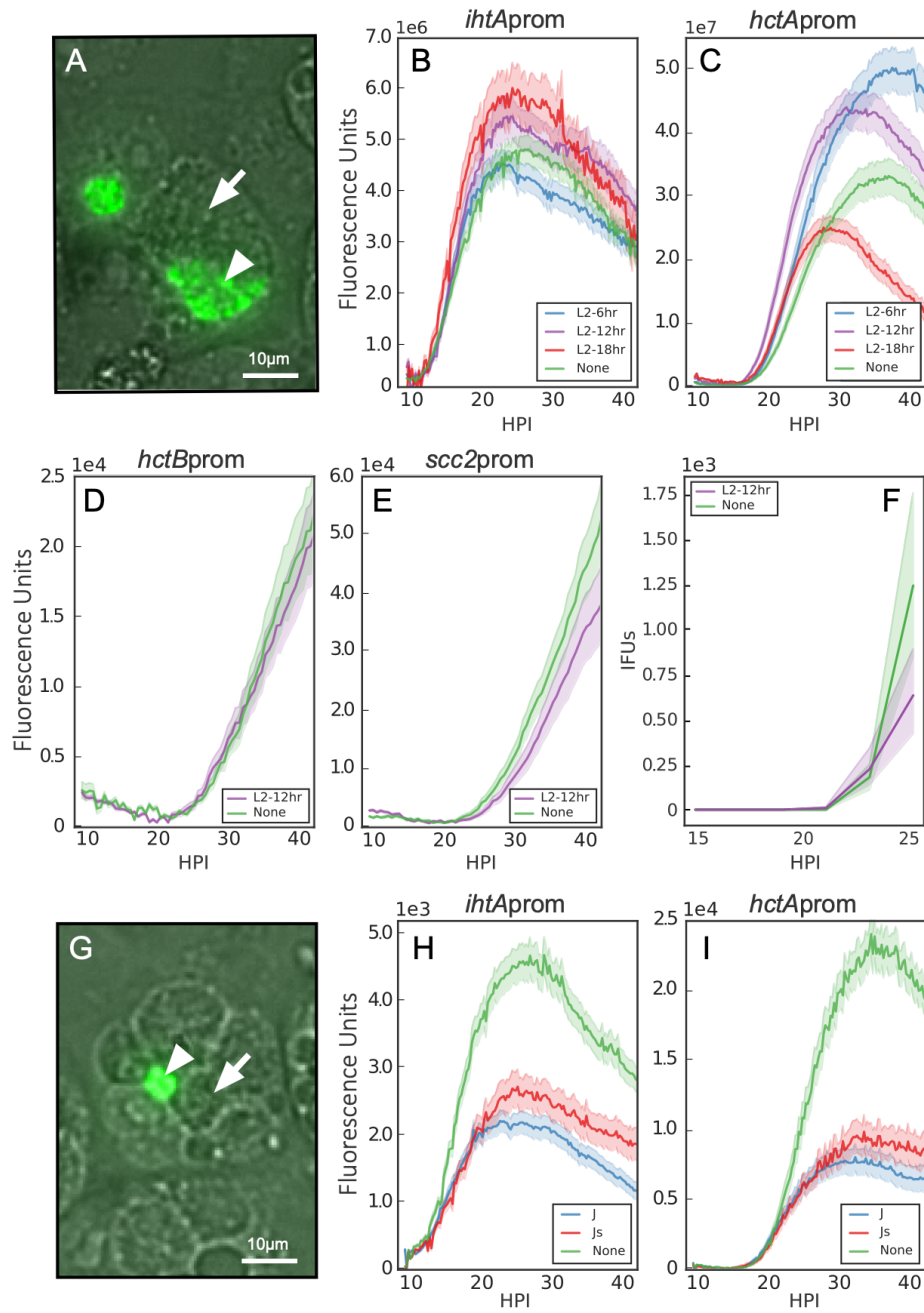
**Figure 2.5: Superinfection does not affect RB-to-EB conversion.**

Host cells were infected with nonfluorescent *C. trachomatis* EBs followed by a secondary infection with *Ctr*-L2-prom EBs at 6, 12, or 18 hpi, and the fluorescent output was compared to that of cells that had not been infected with a primary infection (none). Infections were imaged starting at 9 h postinfection with the *Ctr*-L2-prom strains. (A) Live-cell fluorescence/DIC image of 18-h L2 superinfection with *Ctr-hctA*prom-Clover at 20 magnification (30 h after *Ctr-hctA*prom-Clover infection). Fluorescent signals were measured in inclusions containing both GFP-expressing *C. trachomatis* (arrowhead) and nonfluorescent *C. trachomatis* (arrow). Scale bar, 10 µm. (B and C) The averages of *ihtA*prom-EGFP and *hctA*prom-Clover expression intensities from 50 individual inclusions monitored via automated live-cell fluorescence microscopy during no superinfection (none) and 6, 12, and 18 h *C. trachomatis* L2 superinfections. (D and E) The average fluorescent intensities of 50 individual inclusions using *Ctr-hctB*prom-Clover or *Ctr-scc2*prom-Clover measured with no superinfection (none) or 12 h *C. trachomatis* L2 superinfection. (F) EBs were harvested at 2-h intervals from 15 to 25 h after *Ctr*-L2-prom infection and quantified by replating assay. (G) Live-cell fluorescence/DIC image of cells infected with *C. trachomatis* Js and superinfected with *Ctr-hctA*prom-Clover. The image was taken 30 h after *Ctr-hctA*prom-Clover infection at 20 magnification. Fluorescent signals were measured from inclusions in cells that contained both fluorescent *Ctr-hctA*prom-Clover (arrowhead) and unfused nonfluorescent *C. trachomatis* Js (arrow). Scale bar, 10 µm. (H and I) The average fluorescent intensity of 50 individual inclusions containing *ihtA*prom-EGFP and *hctA*prom-Clover measured with no superinfection (none), *C. trachomatis* J, or *C. trachomatis* Js superinfections. The fluorescent unit cloud represents standard error of the mean (SEM) genome copies, and the IFU cloud represents 95% confidence intervals (CI). y axes are denoted in scientific notation.

**Figure 2.6**
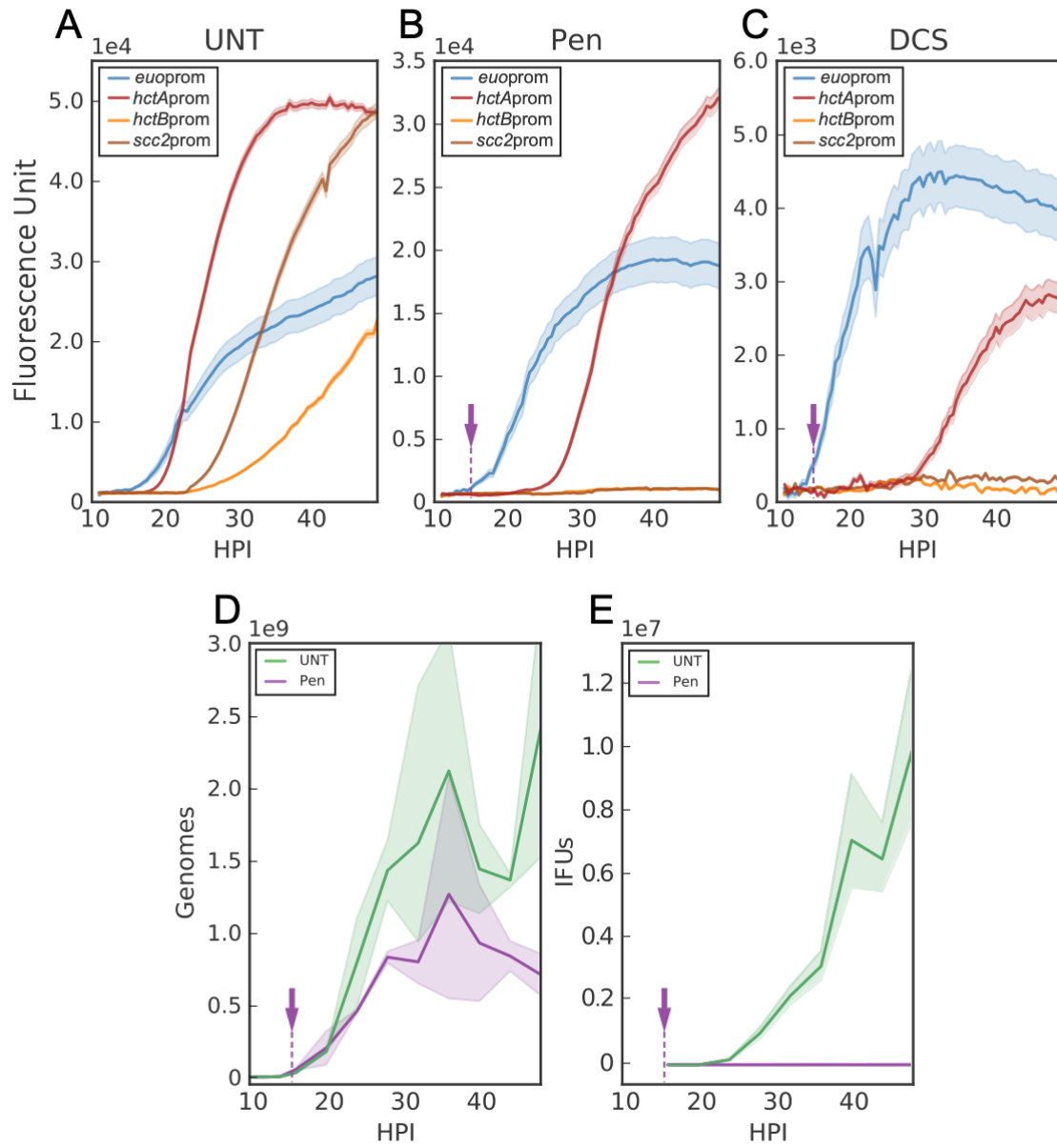
**Figure 2.6: Inhibition of chlamydial cell division inhibits EB conversion.**

Host cells were infected with purified *Ctr*-L2-prom EBs followed by treatment with penicillin G, D-cycloserine, or vehicle only at 14 hpi (purple arrow). (A to C) The averages of RB (*euo*prom-Clover), IB (*hctA*prom-Clover), and EB (*hctB*prom-Clover and *scc2*prom-Clover) expression intensities from 50 individual inclusions monitored via automated live-cell fluorescence microscopy in cells treated with vehicle only (UNT), penicillin (PEN), or D-cycloserine (DCS), respectively. (D) Quantification of genome copy numbers for vehicle only (UNT)- and penicillin (PEN)-treated cells measured using qPCR. (E) Quantification of EB development for vehicle only (UNT)- and penicillin (PEN)-treated cells via replating assay. EBs were harvested at 4-h intervals from 16 to 48 hpi. The fluorescent unit cloud represents standard error of the mean (SEM) genome copies, and the IFU cloud represents 95% confidence intervals (CI). y axes are denoted in scientific notation.

**Figure 2.7**
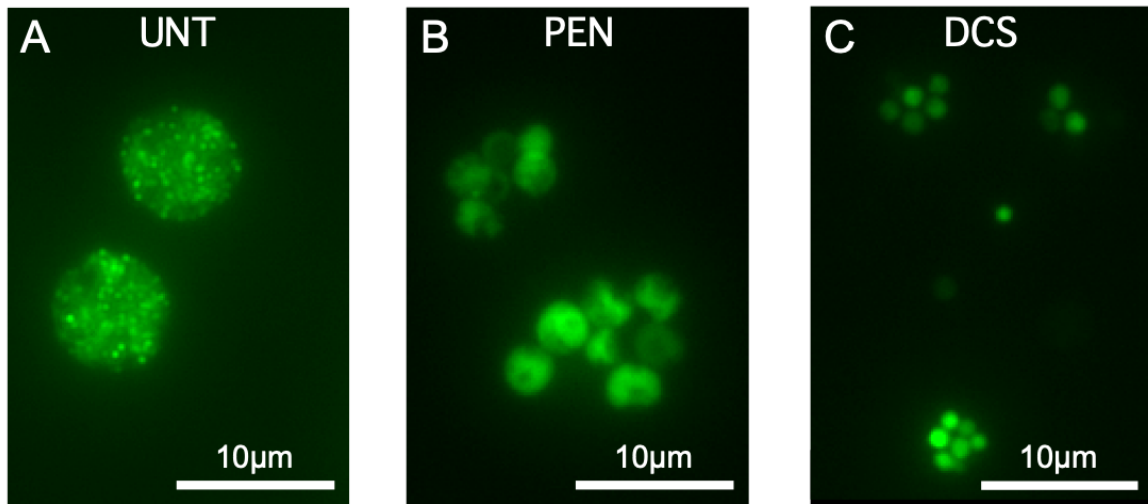


**Figure 2.7: Penicillin G and D-cycloserine induce aberrant cell forms.**

Host cells were infected with *Ctr-hctA*prom-Clover EBs followed by treatment with penicillin or D-cycloserine at 14 hpi. Live-cell fluorescence images were acquired at 40 hpi. (A) Untreated (UNT), vehicle only. (B) Penicillin (PEN) treated. (C) D-cycloserine (DCS) treated. Magnification, 40X. Scale bar, 10 μm.

**Figure 2.8**
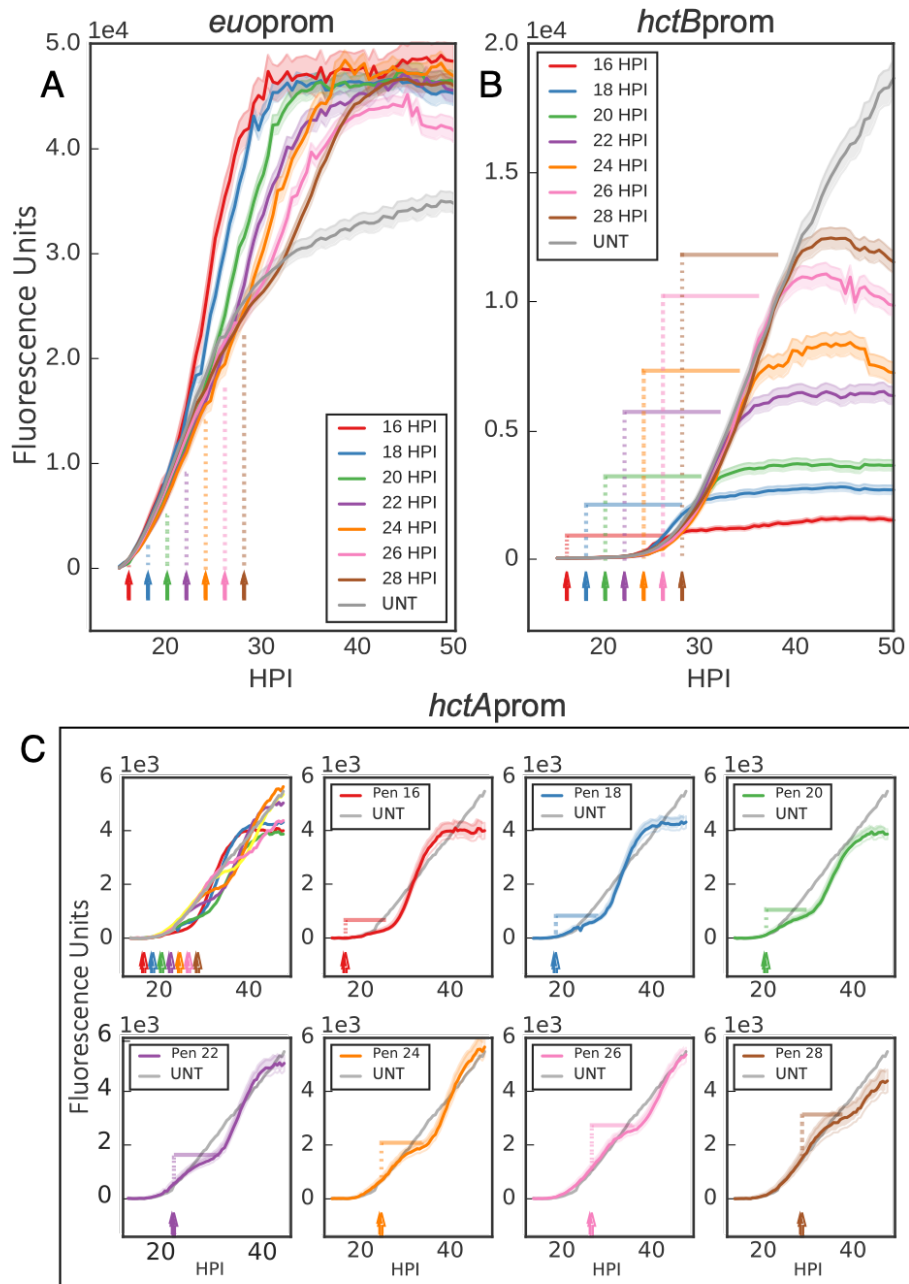
**Figure 2.8: Inhibiting chlamydial cell division inhibits further EB conversion.**

Host cells were infected with *Ctr*-L2-prom EBs followed by treatment with penicillin (Pen) at 2-h intervals starting at 16 hpi or without treatment (UNT). Arrows and vertical dotted lines indicate the addition of penicillin. (A) The averages of *euo*prom-Clover (RB) expression intensities from 50 individual inclusions monitored via automated live-cell fluorescence microscopy for each penicillin treatment (time series starting at 16 hpi) and no treatment (UNT). (B) The averages of *hctB*prom-mKate2 (EB) fluorescence from 50 individual inclusions. Horizontal solid lines indicate time to maximum expression. (C) The averages of *hctA*prom-mKate2 (IB) fluorescence from 50 individual inclusions. *hctA*prom-mKate2 graphs are separated for clarity. Horizontal solid lines indicate time to reinitiation of expression. The cloud represents SEM. y axes are denoted in scientific notation.

**Figure 2.9**
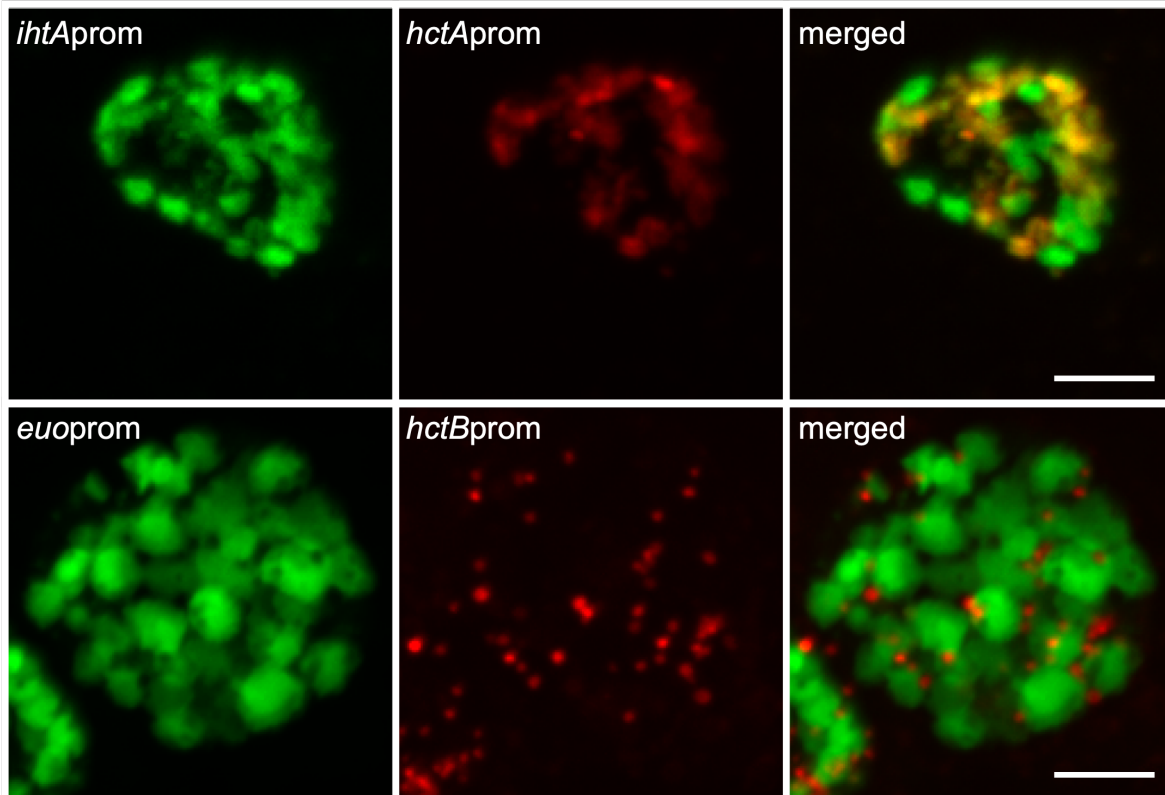


**Figure 2.9: Confocal fluorescence microscopy of cell type promoter expression upon inhibition of chlamydial division.**

Host cells were infected with *Ctr-hctA*prom-mKate2/*ihtA*prom-mNeonGreen (red and green, respectively) or *Ctr-hctB*prom-mKate2/*euo*prom-Clover (red and green, respectively), followed by treatment with penicillin (Pen) at 20 hpi. Samples were fixed at 24 hpi. Fixed samples were imaged by confocal microscopy, and maximum intensity projections are shown. Scale bars, 5 µm.
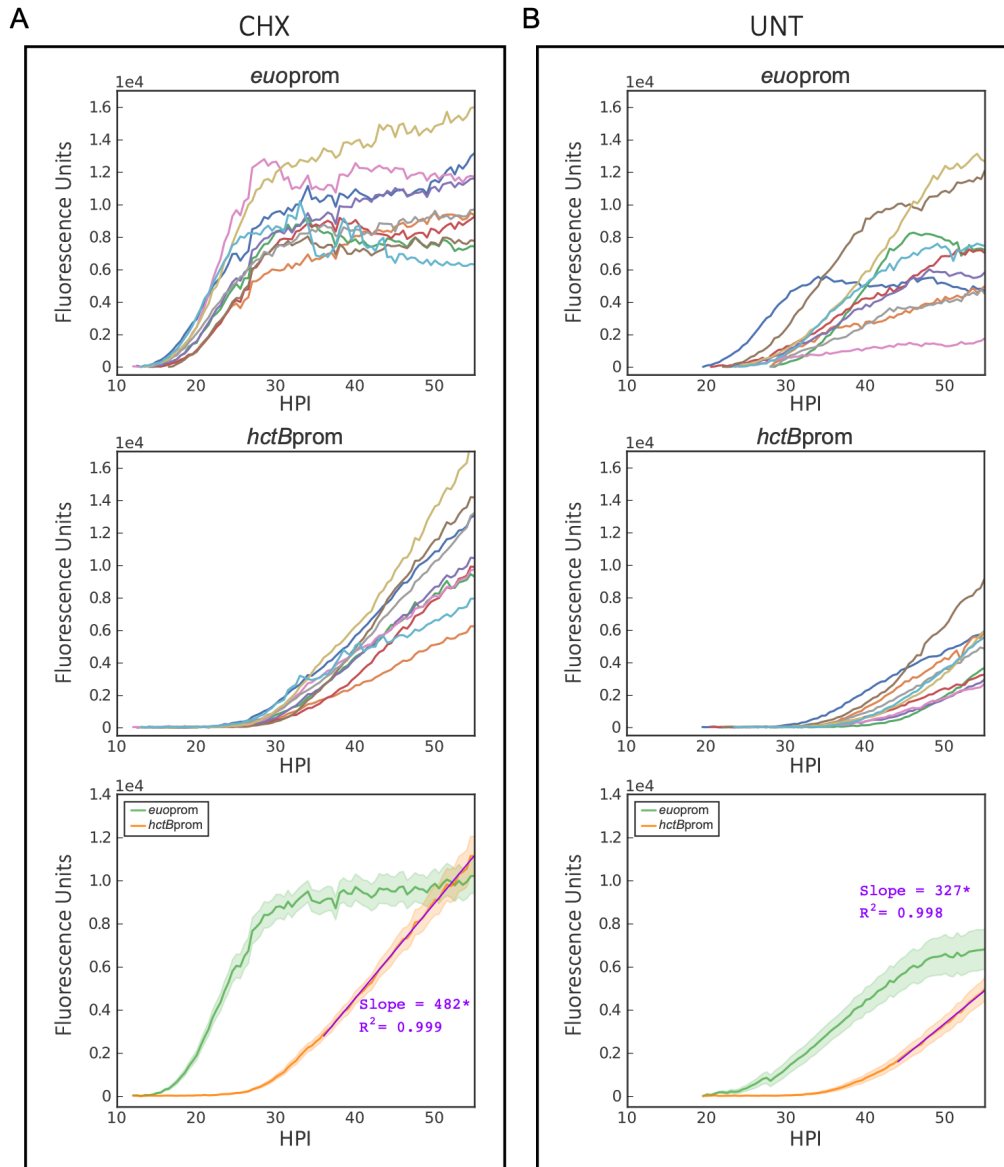
**Figure 2.10**

**Figure 2.10: Effect of cycloheximide on growth rate and EB production.**

Cos-7 cells were either treated with cycloheximide or vehicle only upon infection with *Ctr-hctB*prom-mKate2/*euo*prom-Clover. (A) Individual inclusion traces and averages of *euo*prom-Clover (RB) and *hctB*prom-mKate2 (EB) expression intensities monitored via automated live-cell fluorescence microscopy for cycloheximide (CHX)-treated infections. (B) Individual inclusion traces and averages of *euo*prom-Clover (RB) and *hctB*prom-mKate2 (EB) expression intensities monitored via automated live-cell fluorescence microscopy for vehicle (UNT)-treated infections. Purple lines are linear regression fits. Asterisks denote P value of 0.05. The cloud represents SEM. y axes are denoted in scientific notation.

**Figure 2.11**
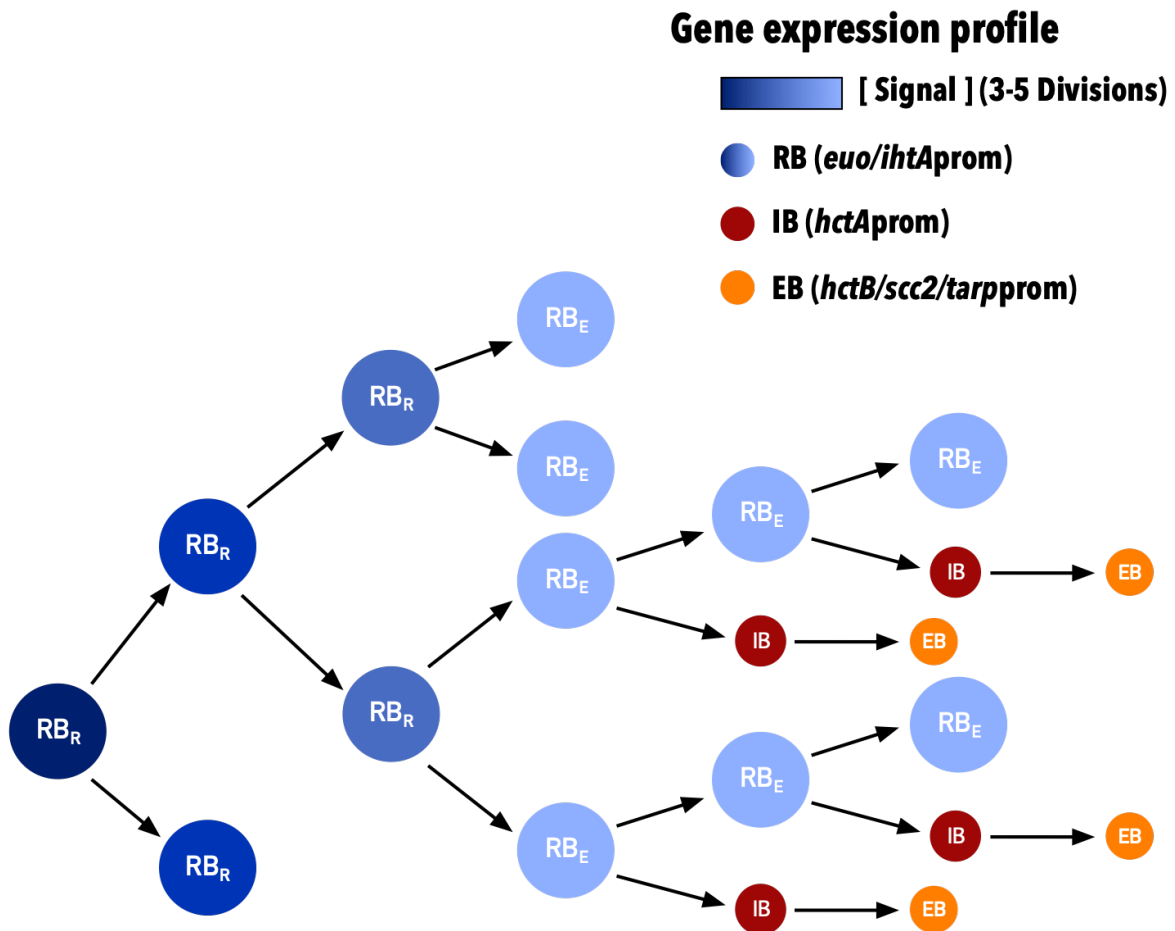


**Figure 2.11: Schematic of concentration-dependent RB/EB conversion model.**
The schematic shows diminishing signal concentration within RBs (dark to light blue) upon cell division. Depletion of the signal permits RBs to produce IBs (red), which then convert to EBs (orange). RBRs divide into two subsequent RBs. RBEs are competent to make EBs and divide into a RB and an IB. Each cell form has predicted associated promoter expression phenotypes. RB (RBR and RBE), *euo-ihtA*; IB, *hctA*; EB, *hctB-scc2-tarp*.

**Figure 2.S1**
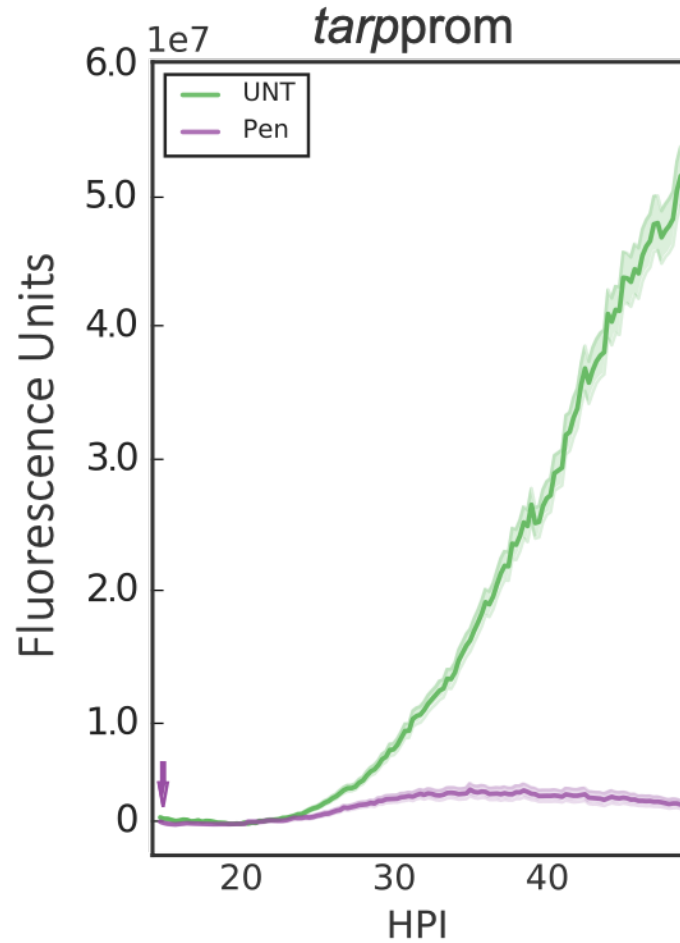


**Figure 2.S1: Inhibition of chlamydial division inhibits tarpprom expression.**
Host cells were infected with purified *Ctr-tarp*prom-Clover EBs followed by penicillin treatment at 14 hpi (arrow). Shown is the average of *tarp*prom-Clover (EB) expression intensities from >50 individual inclusions monitored via automated live-cell fluorescence microscopy in the absence (UNT) and presence of penicillin (Pen). The cloud represents SEM. The y axis is denoted in scientific notation.

**Figure 2.S2**
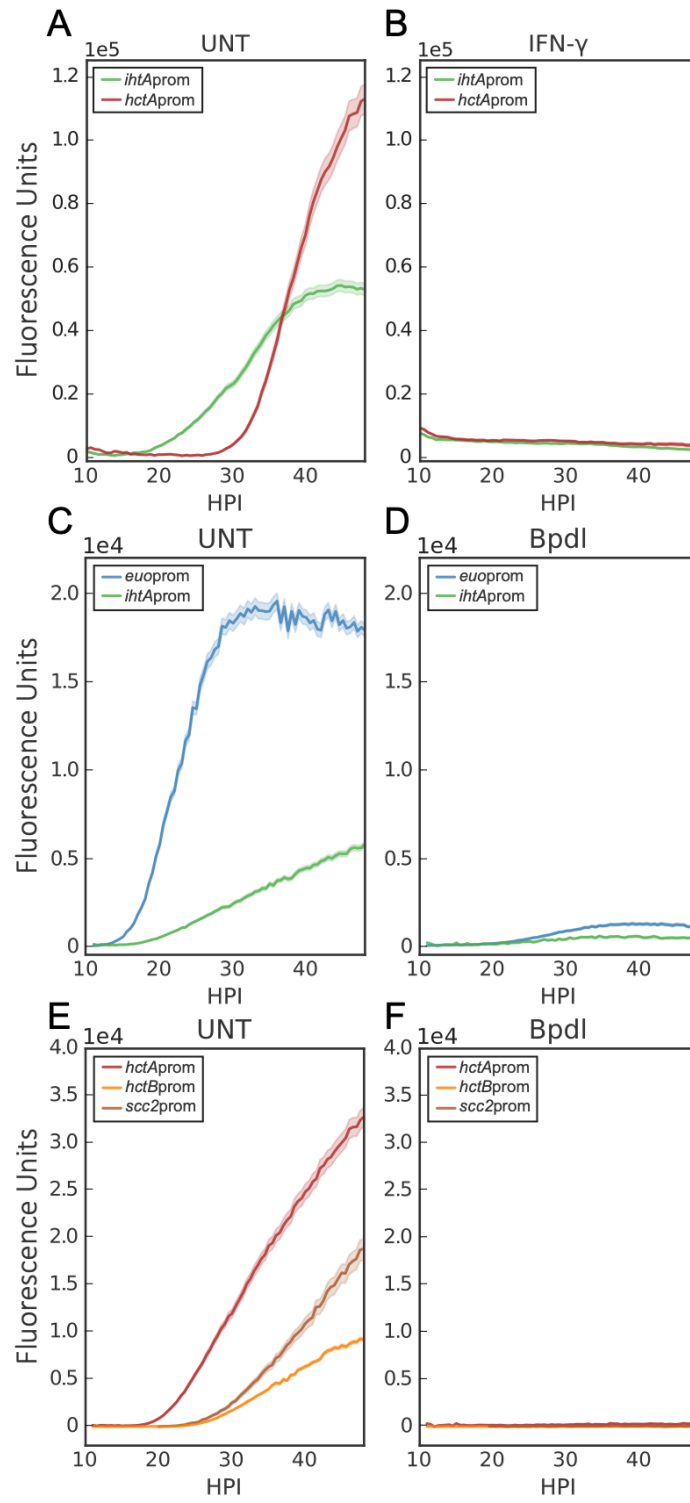
**Figure 2.S2: Effect of interferon gamma and bipyridyl on RB-to-EB conversion.**
HeLa cells were treated with interferon gamma (IFN-γ) 24 h prior to infection with
*Ctr*-L2-prom EBs. Cos-7 cells were treated with bipyridyl upon infection with *Ctr*-L2-
prom EBs. (A and B) The averages of *ihtA*prom-EGFP and *hctA*prom-Clover
expression intensities from >50 individual inclusions monitored via automated live-
cell fluorescence microscopy in the absence (UNT) and presence of IFN-γ. (C and
D) The averages of *ihtA*prom-mNeonGreen and *euo*prom-Clover expression
intensities from >50 individual inclusions monitored via automated live-cell
fluorescence microscopy in the absence (UNT) and presence of bipyridyl (bpdl). (E
and F) The averages of *hctA*prom-mKate2, *hctB*prom-mKate2, and *scc2*prom-Clover
expression intensities from >50 individual inclusions monitored via automated live-
cell fluorescence microscopy in the absence (UNT) and presence of bipyridyl (Bpdl).
The cloud represents SEM. The y axes are denoted in scientific notation.

**Figure 2.S3**
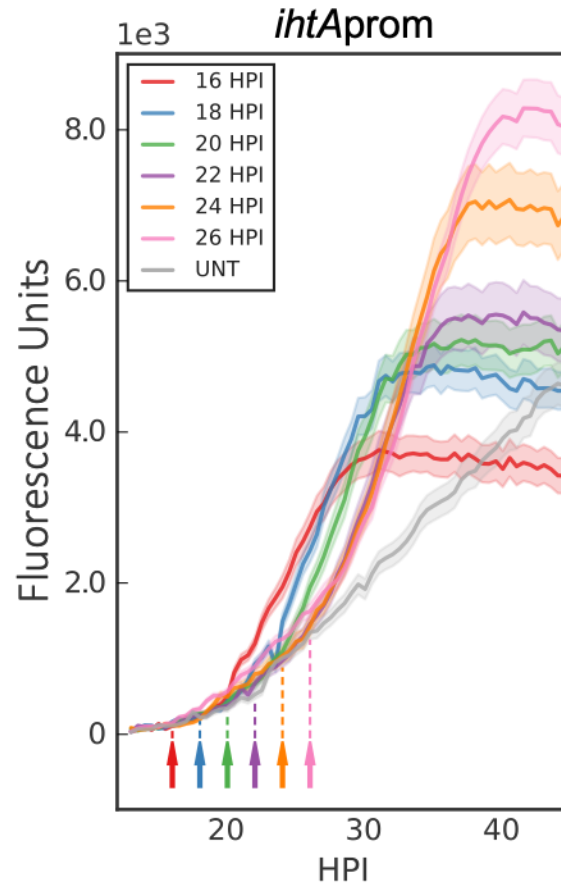


**Figure 2.S3: *ihtA*prom expression increases upon inhibiting cell division.**

Host cells were infected with *Ctr-ihtA*prom-mNeonGreen EBs followed by treatment with penicillin (Pen) at 2-h intervals starting at 16 hpi or without treatment (UNT). Arrows and vertical dotted lines indicate the addition of penicillin. Shown are average *ihtA*prom-mNeonGreen (RB) expression intensities from >50 individual inclusions monitored via automated live-cell fluorescence microscopy for each penicillin treatment (time series starting at 16 hpi) and untreated cells (UNT). The cloud represents SEM. The *y* axis is denoted in scientific notation.

**Figure 2.S4**
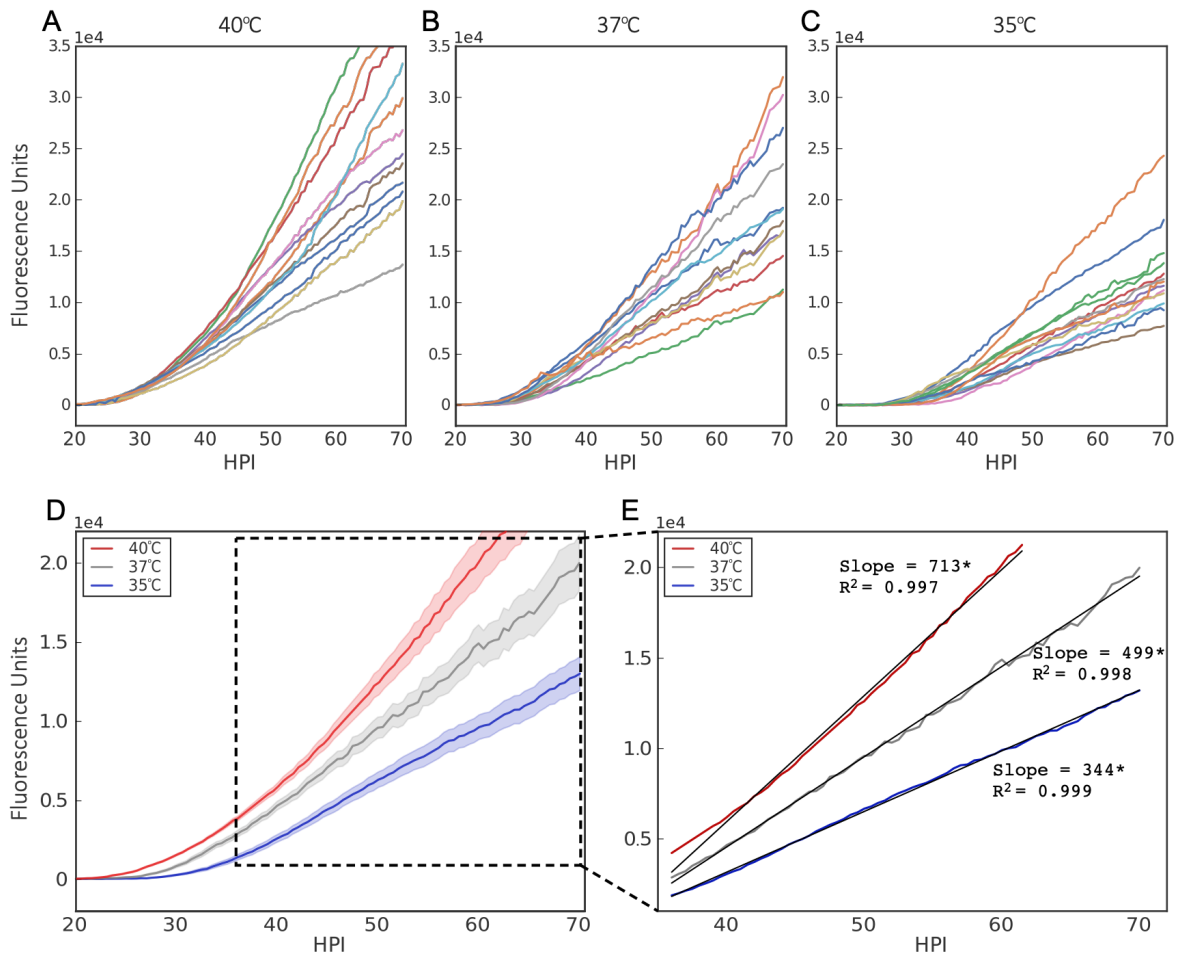


**Figure 2.S4: EB expression follows a linear trajectory late in development.**

Infections with *Ctr*-L2-prom EBs were grown at 35°C, 37°C, and 40°C. (A to C) Individual inclusion traces of *hctB*prom-mKate2 (EB) expression intensities at 35°C, 37°C, and 40°C monitored from 20 to 70 hpi via automated live-cell fluorescence microscopy. (D) The averages of *hctB*prom-mKate2 (EB) expression from the individual inclusion traces for each treatment. (E) *hctB*prom-mKate2 average expression slopes from the three temperatures. Black lines are of *hctB*prom-mKate2 expression for each treatment fit to a linear regression model. Asterisks denote a *P* value of <0.05. The cloud represents SEM. The *y* axes are denoted in scientific notation.

**Movie 2.S1: Live-cell time-lapse movie of inclusion development and *hctA*prom-Clover expression.**

Host cells were infected with purified *Ctr-hctA*prom-Clover EBs. Automated live-cell DIC and fluorescence microscopy were used to capture images every 30 min from 10 to 48 hpi. Download Movie S1, MOV file, 4.8 MB.

**Movie 2.S2: Live-cell time-lapse movie of single inclusion tracking.**

Host cells were infected with the chlamydial transfomant *Ctr-hctB*prom-mKate2/*euo*prom-Clover, and fluorescent images were captured in both the green (Clover) channels as well as the red (mKate2) channels every 30 min, starting at 10 h postinfection, for 55 h. Individual inclusions were identified and tracked using the ImageJ plug-in Trackmate. The fluorescent intensities for each channel were graphed using python and jupyter notebook. The inclusions expressed the green reporter early until peak intensity at ~24 hpi. *hctB*prom was expressed starting at ~22 hpi and continued to increase linearly until inclusion/cell lysis or the experiment was ended. Download Movie S2, MOV file, 0.6 MB.

**Supplemental Material 2.S1: Description of Mathematical Models**

**Model Description**

    We developed two systems of ordinary differential equations to describe the temporal dynamics of reticulate body (RB) to elementary body (EB) conversion. Both models track RB ($R$) replication, their conversion to intermediate bodies (IB, $I$), and subsequent conversion to EBs ($E$). In both models, a signal ($S$) acts as an inhibitor of RB to IB conversion. The models differ in the location of the signal. In the environmental signal model, the signal is located outside of the RB and acts globally on the entire population of Chlamydia while in the intrinsic signal model the signal is located within each individual RB.

    In both models, we assume that cells grow exponentially at per capita rate $r$. The rate of RB conversion to IB follows an inverse Hill function that depends on the concentration of the signal $S$. IBs convert to EBs at per capita rate $\delta$. The signal $S$ is depleted by RBs $R$ at a rate proportional to the RB population. In a single population of cells (multiplicity of infection = 1), this leads to the following system of differential equations for both models

$$\frac{dR}{dt} = rR - \frac{\beta R}{(S/\kappa)^n + 1}$$
$$\frac{dI}{dt} = \frac{\beta R}{(S/\kappa)^n + 1} - \delta I$$
$$\frac{dE}{dt} = \delta I$$
$$\frac{dS}{dt} = -\gamma SR.$$

    Parameters for the model are given in Table 1.

| Parameter | Description | Value |
|:---:|:---|:---:|
| $r$ | RB growth rate | 0.31 hour$^{-1}$ |
| $\beta$ | maximum per capita RB to IB conversion rate | 2.5 hour$^{-1}$ |
| $n$ | Hill coefficient | 2 |
| $\kappa$ | Half-saturation constant | $1/10S_0$ |
| $\delta$ | per capita IB to EB conversion rate | 1 hour$^{-1}$ |
| $\gamma$ | Signal depletion rate | 0.005 cell$^{-1}$ hour$^{-1}$ |

Table 1: Description of parameters used in the mathematical models.

**Multiple infections and superinfections**

The models differ if host cells are multiply infected or superinfected. In each of these cases, the RBs can be divided into subpopulations $R_i$ for i = 1, ..., m that represent RBs derived from each infection. As previously mentioned, in the environmental signal model the signal $S$ is shared by the entire RB population. As such, the governing equations are

$$\frac{dR_i}{dt} = rR_i - \frac{\beta R_i}{(S/\kappa)^n + 1}$$
$$\frac{dI_i}{dt} = \frac{\beta R_i}{(S/\kappa)^n + 1} - \delta I_i$$
$$\frac{dE_i}{dt} = \delta I_i$$
$$\frac{dS}{dt} = -\gamma S \sum_{i=1}^{m} R_i.$$

(1)

In contrast, in the intrinsic model the signal S is unique to each RB so that the governing equations are

$$\frac{dR_i}{dt} = rR_i - \frac{\beta R_i}{(S_i/\kappa)^n + 1}$$
$$\frac{dI_i}{dt} = \frac{\beta R_i}{(S_i/\kappa)^n + 1} - \delta I_i$$
$$\frac{dE_i}{dt} = \delta I_i$$
$$\frac{dS_i}{dt} = -\gamma S_i R_i.$$

(2)

**Differentiating between the environmental and intrinsic models**

To determine conditions in which the temporal dynamics of RBs and EBs differ in the environmental and intrinsic models, we numerically solved models (1) and (2) using the odeint function from the SciPy package under a variety of scenarios. Specifically, we considered variations in multiplicity of infection (MOI), growth rate, and superinfection.

**Parameters and initial conditions for base model**

Numerical solutions were initiated at 8 hours post infection (HPI) and continued until 48 HPI. The RB generation time was calculated to be at max 2.27 hrs as

measured by the increase in *ihtA*prom-EGFP fluorescence using the Growthcurver package in R[47].

By definition, MOI = 1 equates to an average of one EB infecting each host cell. This scenario was used as the baseline for both the environmental and intrinsic models (eqns. (1) and (2)). For these systems of differential equations an MOI = 1 translates into an initial RB population of 1, and as cell-type conversion has yet to occur, the initial IB and EB populations are set to 0.

**Growth rate**

To test the effect of growth rate on each model we implemented the initial conditions from the MOI = 1 scenario, but varied the RB generation time ($r$).

**MOI**

To test the effect of MOI on each model, the initial RB, IB, and EB cell-types were divided into 10 subpopulations. The simulations differed in that the environmental model contained a global signal concentration $S$ shared by all RB subpopulations (eqns. (1)), where the intrinsic model contained 10 independent signal subpopulations $S_i$ which corresponded to each RB subpopulation (eqns. (2)).

**Superinfection**

To test the effect of superinfection on each model we devised two scenarios. For the environmental model (eqns. (1)) we simulated a primary infection at an MOI = 1 saving the 18 HPI RBs, IBs, EBs, and signal concentration values. We then simulated a secondary infection (MOI = 1) and used the 18 HPI signal concentration as $S_0$. The primary infection RB, IB, and EB values were input as a subpopulation in the secondary infection. For the intrinsic model (eqns. (2)), the secondary infection was introduced, but with a self contained signal concentration independent of the primary infection.

**Supplemental Material 2.S2: Inclusion Counting Headless**

```python
1  import os
2  import sys
3  import time
4  from decimal import Decimal
5  from java.awt import Color
6  from java.awt.event import TextListener
7  from ij import IJ
8  from ij import Menus
9  from ij.gui import GenericDialog
10 from ij.io import OpenDialog
11 from ij.measure import ResultsTable
12 from ij.gui import WaitForUserDialog
13 from ij.plugin import ChannelSplitter
14 from ij.plugin import ImageCalculator
15 from net.imglib2.img.display.imagej import ImageJFunctions
16 from java.awt.event import TextListener
17 from ij.measure import ResultsTable
18 import java.util.ArrayList as ArrayList
19 import csv
20 from ij import ImagePlus
21
22 od = OpenDialog("Titer_files", "")
23 firstDir = od.getDirectory()
24 fileList = os.listdir(firstDir)
25
26 if "DisplaySettings.json" in fileList:
27     fileList.remove("DisplaySettings.json")
28 if ".DS_Store" in fileList:
29     fileList.remove(".DS_Store")
30 global fileName
31
32 myfile =
…  open('/home/rickettsia/Desktop/data/Titering/Fancy_Tet_titer/data_2_csv/
…  HctB_48H_with_atc_2.csv', 'wb')
33 wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
34 wr.writerow(["well", "position", "inclusion_num", "DAPI", "DAPI_STD"])
35
36 IJ.run("Set Measurements...", "area mean standard min kurtosis
…  redirect=None decimal=3")
37
38 totalCount = []
39 i = 1
40 fileList.sort()
41 for fileName in fileList:
42     #IJ.run("Collect Garbage")
43     #ip = IJ.getImage()
44     currentFile = firstDir + fileName
45     print("")
```

81

/Users/…/…/…/dissertation_scripts_and_markdown/Inclulsion Counting headless with DA   Page 2/3
Saved: 5/5/22, 10:29:43 AM                                          Printed for: Travis

```python
46      print(currentFile)
47      ip = IJ.openImage(currentFile)
48      #ip.show()
49      fileName = ip.title
50      IJ.run(ip, "Set Scale...", "distance=0")
51      IJ.run(ip, "Gaussian Blur...", "sigma=3")
52      IJ.run(ip, "Unsharp Mask...", "radius=4 mask=0.60")
53      IJ.run(ip, "Subtract Background...", "rolling=50")
54      IJ.run(ip, "Enhance Contrast...", "saturated=0.3")
55      #imp.setRoi(318, 246, 1581, 1647)
56      IJ.setAutoThreshold(ip, "MaxEntropy dark")
57      IJ.run(ip, "Measure", "")
58      IJ.resetThreshold(ip)
59      rt = ResultsTable()
60      rt = rt.getResultsTable()
61      DAPI_sig = rt.getValueAsDouble(1, 0)
62      D_kurt = rt.getValueAsDouble(2, 0) #standard deviation
63      IJ.run("Clear Results")
64
65      channels = ChannelSplitter.split(ip)
66      imp_DAPI = channels[0]
67      imp_GFP = channels[1]
68      #IJ.selectWindow(fileName)
69      #IJ.run("Close")
70      #imp_GFP.show()
71
72      IJ.setThreshold(imp_GFP, 1111, 65536)
73      IJ.run(imp_GFP, "Make Binary", "")
74      IJ.run(imp_GFP, "Fill Holes", "")
75      IJ.run(imp_GFP, "Watershed", "")
76      IJ.run("Clear Results")
77
78      IJ.run(imp_GFP, "Analyze Particles...", "size=200-2000
…  circularity=0.25-1.00 show=Nothing display include")
79      print(fileName)
80      rt = ResultsTable()
81      rt = rt.getResultsTable()
82      numInclusions = rt.getCounter()
83      print('inclusions counted')
84      print(numInclusions)
85      well = fileName.split('_')[5]
86      well = well.split('-')[0]
87      position = fileName.split('_')[6]
88      position = position.split('.')[0]
89      print(well)
90      print(position)
91      l1 = (well, position, numInclusions, DAPI_sig,  D_kurt)
92      wr.writerow(l1)
```

```
93     print(l1)
94     #wr.writerow('')
95     IJ.run("Clear Results")
96     i = i+1
97     print(i)
98     #IJ.selectWindow('C2-'+fileName)
99     #IJ.run("Close")
100
101 myfile.close()
```

**Table 2.1**

| Construct | Primer Name | Template |
|---|---|---|
| **p2TK2-*iht*Aprom-EGFP** | | |
| GgtacCTAGAATTAAAGAGGAGAAATTAAGCATGCGTAA AGGAGAAGAACTTTTCACTGGAGTTGTCCCAATTCTT GTTGAATTAGATGGTGATGTTAATGGGCACAAATTTTC TGTCAGTGGAGAGGGTGAAGGTGATGCAACATACGG AAAAACTTACCCTTAAATTTATTTGCACTACTGGAAAACT ACCTGTTCCATGGCCAACACTTGTCACTACTTTCGGTT ATGGTGTTCAATGCTTTGCGAGATACCCAGATCATATG AAACAGCATGACTTTTTCAAGAGTGCCATGCCCGAAG GTTATGTACAGGAAAGAACTATATTTTTCAAAGATGAC GGGAACTACAAGACACGTGCTGAAGTCAAGTTTGAAG GTGATACCCTTGTTAATAGAATCGAGTTAAAAGGTATT GATTTTAAAGAAGATGGAAACATTCTTGGACACAAATT GGAATACAACTATAACTCACACAATGTATACATCATGGC AGACAAACAAAAGAATGGAATCAAAGTTAACTTCAAAA TTAGACACAACATTGAAGATGGAAGCGTTCAACTAGC AGACCATTATCAACAAAATACTCCAATTGGCGATGGCC CTGTCCTTTTACCAGACAACCATTACCTGTCCACACAA TCTGCCCTTTCGAAAGATCCCAACGAAAAGAGAGACC ACATGGTCCTTCTTGAGTTTGTAACAGCTGCTGGGATT ACACATGGCATGGATGAACTATACAAATAGGGATGACA TGTGATTCGCGTAGGAAAAAGAGGAGGGAGACCTCC TCTTTTTTTTTATTTTGTAGAGTTCCGTTACTATTGGCA CCCTGTGTGCAGTTAGGATGAGTAGACTAGTTCTGCA GCCTTTTACAGGGTGTTATGTTTTGCATTGCAAAAAGC TCCTAAGACGCGGCCGCGTCGACGGATCCCTTGTAC | EGFP gblock | |
| TATTACAAccctcgtcacgcccctgaatgccagc | 5' *iht*Aprom | **L2 Genomic** |
| CGCTCACCATGCTTAATTTCTCCTCTTTAATTCTAGgtac Cgcaactcttataacattattccgc | 3' *iht*Aprom | **L2 Genomic** |
| GACGGATCCCTTGTACAATCAATTTACCGATTAAATAGT CTCTATAATTCAC | 5' p2tK2-sw2 vector EGFP | **p2TK2SW2** |
| ttcaggggcgtgacgagggTTGTAATACGGTTATCCACAG | 3' p2tK2-sw2 vector ihtAprom | **p2TK2SW2** |
| **p2TK2SW2 ampR to specR** | | |
| atgcgctcacgcaactggtccagaACCTTGACCGAACGCAGCG GTG | 5' *aadA* (spec) from pBam4 | pBam4 |
| ttggtctgacagTTATTTGCCGACTACCTTGGTGATCTCG | 3' *aadA* (spec) from pBam4 | pBam4 |
| gtagtcggcaaaTAACTGTCAGACCAAGTTTACTCATATATA C | 5' p2tK2-sw2 vector | **p2TK2-*iht*Aprom-EGFP** |

| | | |
|---|---|---|
| tctggaccagttgcgtgagcgcatCATTGGAAAACGTTCTTCGGGGCGAAAAC | 3' p2tK2-sw2 vector | **p2TK2-*ihtA*prom-EGFP** |
| **p2TK2-*hctA*prom-Clover** | | |
| GGCAAAAAAAATGgaattcggcatggtgagcaagggcgaggagctgttcaccggggtggtgcccatcctggtcgagctggacggcgacgtaaacggccacaagttcagcgtccgcggcgagggcgagggcgatgccaccaacggcaagctgaccctgaagttcatctgcaccaccggcaagctgcccgtgccctggcccaccctcgtgaccaccttcggctacggcgtggcctgcttcagccgctaccccgaccacatgaagcagcacgacttcttcaagtccgccatgcccgaaggctacgtccaggagcgcaccatctctttcaaggacgacggcacctacaagacccgcgccgaggtgaagttcgagggcgacaccctggtgaaccgcatcgagctgaagggcatcgacttcaaggaggacggcaacatcctggggcacaagctggagtacaacttcaacagccacaacgtctatatcacggccgacaagcagaagaacggcatcaaggctaacttcaagatccgccacaacgttgaggacggcagcgtgcagctcgccgaccactaccagcagaacacccccatcggcgacggccccgtgctgctgcccgacaaccactacctgagccatcagtccgccctgagcaaagaccccaacgagaagcgcgatcacatggtcctgctggagttcgtgaccgccgccgggattacacatggcatggacgagctgtacaagTAGGGATGACATGTGATTCGC | Clover gblock | |
| CCGTATTACAtttagattctagaaaatggttgcatgaatttg | 5' *hctA*prom Clover | **L2 Genomic** |
| ttgctcaccatgccgaattcCATTTTTTTTGCCGTATCTTTTAGCGCCATg | 3' *hctA*prom Clover | **L2 Genomic** |
| TAGGGATGACATGTGATTCGCG | 5' EGFP vector *hctA*prom | **p2TK2-*ihtA*prom-EGFP** |
| ccattttctagaatctaaaTGTAATACGGTTATCCACAG | 3' EGFP vector *hctA*prom | **p2TK2-*ihtA*prom-EGFP** |
| **p2TK2-*hctB*prom-Clover** | | |
| CCGTATTACATgttaaaaactaaccatttttattaaagtttttcattctccttgtcgat | 5' *hctB*prom Clover | **L2 Genomic** |
| ttgctcaccatgccgaattcGCGTTTCTTTTGTACTCCCAACATGTTC | 3' *hctB*prom Clover | **L2 Genomic** |
| cgaattcggcatggtgagcaagggcgaggagctgttcacc | 5' Clover vector *hctB*prom | **p2tK2-sw2-*hctA*prom-Clover** |
| gtttttaacATGTAATACGGTTATCCACAGAATCAGGGGATAACGCAGG | 3' Clover vector *hctB*prom | **p2tK2-sw2-*hctA*prom-Clover** |
| **p2TK2-*tarp*prom-Clover** | | |
| ccgtattacaTATTGCATTTCTTCACAAACGTTACCCGG | 5' *tarp*prom Clover | **L2 Genomic** |
| tgctcaccatAGAATTCGTCATAACTACAAATTAAATAAAAACAAC | 3' *tarp*prom Clover | **L2 Genomic** |
| gacgaattctATGGTGAGCAAGGGCGAGGAGCTG | 5' Clover vector *tarp*prom | **p2tK2-sw2-*hctA*prom-Clover** |
| aaatgcaataTGTAATACGGTTATCCACAGAATCAGG | 3' Clover vector *tarp*prom | **p2tK2-sw2-*hctA*prom-Clover** |

| | | |
|---|---|---|
| **p2TK2-*scc2*prom-Clover** | | |
| CCGTATTACAcgatgttaacttacgcaaaaagaattagttatg | 5' *scc2*prom Clover | **L2 Genomic** |
| TGCCGAATTCtttagaattattagaagatggagtgctcatc | 3' *scc2*prom Clover | **L2 Genomic** |
| taattctaaagaattcggcatggtgagcaaggg | 5' Clover vector *scc2* | **p2tK2-sw2-*hctA*prom-Clover** |
| gttaacatcgTGTAATACGGTTATCCACAGAATCAGG | 3' Clover vector *scc2* | **p2tK2-sw2-*hctA*prom-Clover** |
| **p2TK2-*euo*prom-Clover** | | |
| CCGTATTACAtatttttaacaaaccacttgattaataagtttttg | 5' *euo*prom Clover | **L2 Genomic** |
| TGCCGAATTCgacccctgtatcttgttgtaagcattcc | 3' *euo*prom Clover | **L2 Genomic** |
| TACAGGGGTCgaattcggcatggtgagcaaggg | 5' Clover vector *euo*prom | **p2tK2-sw2-*hctA*prom-Clover** |
| gttaaaaataTGTAATACGGTTATCCACAGAATCAGG | 3' Clover vector *euo*prom | **p2tK2-sw2-*hctA*prom-Clover** |
| **p2TK2-*hctA*prom-mKate2** | | |
| AAAAATGgcggatcCGATGGTGAGCGAGCTGATTAAGGAGAACATGCACATGAAGCTGTACATGGAGGGCACCGTGAACAACCACCACTTCAAGTGCACATCCGAGGGCGAAGGCAAGCCCTACGAGGGCACCCAGACCATGAGAATCAAGGCGGTCGAGGGCGGCCCTCTCCCCTTCGCCTTCGACATCCTGGCTACCAGCTTCATGTACGGCAGCAAAACCTTCATCAACCACACCCAGGGCATCCCCGACTTCTTTAAGCAGTCCTTCCCCGAGGGCTTCACATGGGAGAGAGTCACCACATACGAAGACGGGGGCGTGCTGACCGCTACCCAGGACACCAGCCTCCAGGACGGCTGCCTCATCTACAACGTCAAGATCAGAGGGGTGAACTTCCCATCCAACGGCCCTGTGATGCAGAAGAAAACACTCGGCTGGGAGGCCTCCACCGAGACCCTGTACCCCGCTGACGGCGGCCTGGAAGGCAGAGCCGACATGGCCCTGAAGCTCGTGGGCGGGGGCCACCTGATCTGCAACTTGAAGACCACATACAGATCCAAGAAACCCGCTAAGAACCTCAAGATGCCCGGCGTCTACTATGTGGACAGAAGACTGGAAAGAATCAAGGAGGCCGACAAAGAGACCTACGTCGAGCAGCACGAGGTGGCTGTGGCCAGATACTGCGACCTCCCTAGCAAACTGGGGCACAGATAGGGATGACATGTGA | mKate2 gBlock | |
| TAGGGATGACATGTGATTCGCGTAGGAAAAAGAGGAGG | 5' replace Clover with mKate2 | **p2tK2-sw2-*hctA*prom-Clover** |
| CGgatccgcCATTTTTTTTGCCGTATCTTTTAGCG | 3' replace Clover with mKate2 | **p2tK2-sw2-*hctA*prom-Clover** |
| **p2TK2-*ihtA*prom-EGFP_*hctA*prom-mKate2** | | |

| | | |
|---|---|---|
| GCGGCCGCGTCAccctcgtcacgcccctgaatgc | 5' *iht*Aprom-EGFP | **p2TK2-*iht*Aprom-EGFP** |
| GGGATCCGTCCTATTTGTATAGTTCATCCATGCCATGTC | 3' *iht*Aprom-EGFP | **p2TK2-*iht*Aprom-EGFP** |
| ATACAAATAGGACGGATCCCTTGTACAATCAATTTACCG | 5' *hct*Aprom-mKate2 vector *iht*Aprom-EGFP | **p2TK2-*hct*Aprom-mKate2** |
| tgacgagggTGACGCGGCCGCGTCTTAGGAGC | 3' *hct*Aprom-mKate2 vector *iht*Aprom-EGFP | **p2TK2-*hct*Aprom-mKate2** |
| **p2TK2-*iht*Aprom-mNeonGreen_*hct*Aprom-mKate2** | | |
| TTAAAGAGGAGAAATTAAGCatggtgagcaaaggcgaagaagataacatggcgagcctgccggcgacccatgaactgcatattttttggcagcattaacggcgtggattttgatatggtgggccagggcaccggcaacccgaacgatggctatgaagaactgaacctgaaaagcaccaaaggcgatctgcagtttagcccgtggattctggtgccgcatattggctatggctttcatcagtatctgccgtatccggatggcatgagcccgtttcaggcggcgatggtggatggcagcggctatcaggtgcatcgcaccatgcagtttgaagatggcgcgagcctgaccgtgaactatcgctatacctatgaaggcagccatattaaaggcgaagcgcaggtgaaaggcaccggctttccggcggatggcccggtgatgaccaacagcctgaccgcgcggcggattggtgccgcagcaaaaaaacctatccgaacgataaaaccattattagcacctttaaatggagctataccaccggcaacggcaaacgctatcgcagcaccgcgcgcaccacctataccttttgcgaaaccgatggcggcgaactatctgaaaaaccagccgatgtatgtgtttcgcaaaaccgaactgaaacatagcaaaaccgaactgaactttaaagaatggcagaaagcgtttaccgatgtgatgggcatggatgaactgtataaaTAGGACGGATCCCTTGTACA | mNeonGreen gblock | |
| TAGGACGGATCCCTTGTACAATCAATTTACCGATTAAATAGTCTC | 5' mNG replace EGFP | **p2TK2-*iht*Aprom-EGFP_*hct*Aprom-mKate2** |
| GCTTAATTTCTCCTCTTTAATTCTAGgtacCgcaactcttataacttattcc | 3' mNG replace EGFP | **p2TK2-*iht*Aprom-EGFP_*hct*Aprom-mKate2** |
| **p2TK2-*iht*Aprom-EGFP_*hct*Bprom-mKate2** | | |
| CgaattcggcATGGTGAGCGAGCTGATTAAGGAGAACATGCAC | 5' mKate2 prom vector | **p2TK2-*iht*Aprom-EGFP_*hct*Aprom-mKate2** |
| gtttttaacATGTAATACGGTTATCCACAGAATCAGGGGATAACGCAGG | 3' mKate2 prom vector | **p2TK2-*iht*Aprom-EGFP_*hct*Aprom-mKate2** |
| CCGTATTACATgttaaaaactaaccattttttattaaagttttcattctccttgtcgat | 5' *hct*Bprom (mKate2) | **p2TK2-*hct*Bprom-Clover** |
| CGCTCACCATgccgaattcGCGTTTCTTTTGTACTCCCAACATGTTCATtcc | 3' *hct*Bprom (mKate2) | **p2TK2-*hct*Bprom-Clover** |
| **p2TK2-*euo*prom-Clover_*hct*Bprom-mKate2** | | |
| GCTGTACAAGTAGGACGGATCCCTTGTACAATCAATTTACCGATTAAATAGTCTC | 5' mKate2-EGFP vector | **p2TK2-*iht*Aprom-EGFP_*hct*Bprom-mKate2** |
| gttaaaaataGACGCGGCCGCGTCTTAGGAGCTTTTTGCAATGC | 3' mKate2-EGFP vector | **p2TK2-*iht*Aprom-EGFP_*hct*Bprom-mKate2** |

| | | |
|---|---|---|
| CGGCCGCGTCtatttttaacaaaccacttgattaataagtttttttgttgggaaaatattacc | 5' *euo*prom-Clover | **p2TK2-*euo*prom-Clover** |
| ATCCGTCCTACTTGTACAGCTCGTCCATGCCATGTGTAATCCC | 3' *euo*prom-Clover | **p2TK2-*euo*prom-Clover** |

# CHAPTER THREE: FORMATION OF THE CHLAMYDIAL INTERMEDIATE BODY IS BEST EXPLAINED AS RESULTING FROM ASYMMETRIC REPLICATION

Travis J. Chiarelli[a], Nicole A. Grieshaber[a], Cody Appa[a], Scott S. Grieshaber[a]

[a]Department of Biological Sciences, University of Idaho, Moscow, Idaho, USA

All supplemental movies used in this manuscript can be downloaded from Supplemental_Movies.zip.

**Abstract**

*Chlamydia trachomatis* is an obligate intracellular bacterium that progresses through an essential multi cell form developmental cycle. Infection of the host is initiated by the elementary body (EB). Once in the host, the EB differentiates into the non-infectious, but replication competent, reticulate body, or RB. After multiple rounds of replication, RBs undergo secondary differentiation, producing transitory forms, termed intermediate bodies (IBs), and eventually newly infectious EBs. Our previous study indicated that RB-to-EB differentiation was likely responding to an intrinsic signal, dependent on growth rate and cell division. Using agent-based models, we explored the cell-form dynamics of two IB developmental hypotheses (asymmetric division and direct conversion) and tested the model predictions with our newly modified chlamydial reporter strains. Results from this study showed that, after RB amplification, individual $RB_E$s remain stable throughout the infectious cycle. Live-cell kinetic and single-cell experiments demonstrated that after cell division is inhibited, RBs do not decrease in number, indicating that RBs are not converting directly to IBs. Analysis of the IB and EB reporters also demonstrated that IBs are a transient cell population, and although IB are produced by RBs in a cell division dependent process, they mature directly into EBs post cell division. The culmination of these results suggests that development of IBs is cell division dependent and likely occurs via asymmetrical division from $RB_E$s.

**Introduction**

*Chlamydiae* are obligate intracellular bacterial parasites that cause an array of diseases in both humans and animals. *Chlamydia trachomatis*, a human-adapted species, is the leading global cause of bactrerial sexually acquired infections and preventable blindness. In 2019, the CDC reported 1.8M *C. trachomatis* infections within the United States alone, with the most recent reports indicating that rates increased 10.0% in women and 32.1% in men from 2015-19 [1, 2]. This increase in infection rates has been seen across all racial/ethinic groups and affects all age ranges [2].

Chlamydial growth and development has classically been characterized as a biphasic cycle, consisting of two primary cell forms: the elementary and reticulate body. These cell forms maintain a division of labor throughout the infectious cycle and are essential for chlamydial proliferation. The elementary body (EB) is the infectious cell form and initiates host cell invasion by pathogen-mediated endocytosis [3]. The EB cell form is non-replicative, containing a condensed nucleoid, and remains outside the cell cycle [4, 5]. Upon entry into the host, the EB undergoes large transcriptional and phenotypic changes, maturing into the reticulate body (RB) in a process that takes ~11 h [6, 7]. The RB is the replication competent, or vegatative, cell form. However, the RB is non-infection and must re-differentiate back into the EB for subsequent rounds of infection to occur [8].

Live-cell cell-form specific promoter-reporter expression kinetics from our previously published work demonstrated that IB/EB production is dependent on both the growth rate of *Chlamydia* and cell division. We also showed that EB production was linear and corresponded to a plateau in RB-associated expression late in the developmental cycle (>24 hpi), suggesting that RB-to-EB development may be occurring via asymmetric cell division [9]

In this study, we created new live-cell reporters to more accurately differentiate between each individual cell form (RB, IB, and EB). We used kinetics from these reporters to guide the construction of agent-based models that simulated the chlamydial developmental cycle. To determine the role of cell division in development of the IB, we simulated two developmental mechanisms (RB-to-IB

direct conversion and IB asymmetric production from RBs). The outputs of these simulations indicated that these hypotheses could be differentiated from one another by inhibiting chlamydial cell division. Using our newly developed dual reporter chlamydial strains we show that RB depletion does not occur after inhibition of cell division and that cell division is required for further development of IBs, suggesting that asymmetric division is the likely mechanism for IB production. We also provide direct evidence that IB-to-EB development occurs post cell division by direct maturation and is a cell-division independent process.

**Results**

**Development of a live-cell reporter system to monitor active cell-form specific gene expression.** In our previous study we discovered that inclusion-level cell-form specific development appeared to occur in two stages. The first stage, occurring  between 12-24 hpi, was characterized by an increase in RB-associated expression. This was followed by a plateau in RB expression which corresponded with linear EB-associated reporter production [9]. We hypothesized that these kinetics were due to a developmental program consisting of two primary stages: RB amplification, where immature RBs (deemed $RB_R$s) divide symmetrically to amplify RB numbers, and asymmetric EB production, where mature RBs ($RB_E$s) divide asymmetrically to produce IB (and subsequently EBs).

To further investigate our hypothesized model of development, we created three dual reporter chlamydial strains to visualize active cell-form specific expression at the single-cell level. The RB/IB promoter-reporter was constructed using the promoter of the RB-associated gene *euo* (Early Upstream Open reading frame) [10, 11] and the promoter of *hctA*. In our previous study, we designated *hctA* as an IB-associated promoter as *hctA* exhibited differential regulation in comparison to other "true" late EB genes (i.e. *hctB*, *scc2*, and *tarp*), initiating earlier in development and exhibiting an altered expression profile in division inhibited *Chlamydia* [9]. To monitor active RB expression, the *euo* promoter was used to drive the expression of a photo-stable fluorescent protein variant, mNeonGreen (mNG), with an attached to the LVA protein degradation tag [12, 13]. For IB expression, the *hctA* promoter drove the

photostable red fluorescent protein (mKate2) [14]. The RB/EB and IB/EB dual reporters were constructed in a similar fashion consisting of *euo*prom-mNG(LVA)/ *hctB*prom-mKate2 and *hctA*prom-mNG(LVA)/*hctB*prom-mKate2, respectively. These dual promoter-reporter constructs were transformed into *Chlamydia* L2, creating the *Ctr*-L2-*euo*prom-mNG(LVA)_*hctA*prom-mKate2, *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2, and *Ctr*-L2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2 reporter strains (**Fig. 3.1**).

Automated live-cell microscopy was used to monitor the cell-form specific expression kinetics of each strain within individual inclusions and confocal microscopy was used to image each cell-form subpopulation at the intra-inclusion level. Confocal images of host cells infected with each promoter-reporter strain revealed that cell-form specific fluorescent reporter expression existed in isolated cells, where there was no overlap of the RB (*euo*prom-mNG(LVA)) and IB (*hctA*prom-mKate2) (**Fig. 3.1A**), the RB (*euo*prom-mNG(LVA)) and EB (*hctB*prom-mKate2) (**Fig. 3.1B**), nor the IB (*hctA*prom-mNG(LVA)) and EB (*hctB*prom-mKate2) associated reporters (**Fig. 3.1C**), indicating that active expression from each promoter existed in discrete cell forms. In automated live-cell experiments host cells were infected with each strain and the fluorescent intensity of each promoter-reporter was measured from 10-50 hpi (**Fig. 3.1D**). On average, mNG(LVA) expression driven by the RB-associated promoter *euo* was detected starting at ~11 hours post infection (hpi); expression then plateaued at ~24 hpi and remained level throughout the duration of the experiment. Expression from the IB-associated promoter-reporter, *hctA*prom-mNG(LVA), was detected beginning at ~18 hpi and began to plateau ~10 h later. First detection of the EB promoter-reporter (*hctB*prom-mKate2) was at ~24 hpi and was followed by linear reporter production which continued throughout the remainder of the experiment. The kinetics from the RB and EB reporter strains were consistent with previously published expression data [9]. Whereas, the plateau in active *hctA*prom-mNG(LVA) expression has not been previously reported.

The increase and plateau in expression levels of both the *euo*prom-mNG(LVA) and *hctA*prom-mNG(LVA) reporter strains from the live-cell experiments suggested

the total number of RBs and IBs within each inclusion would reach a maximum value. To verify that these plateaus were in fact due to maximum RB and IB numbers, fixed confocal images of host cells infected with the *euo*prom-mNG(LVA) and *hctA*prom-mNG(LVA) promoter-reporter strains were taken of individual inclusions in 2 h increments throughout the infectious cycle (**Fig. 3.2**). Samples were stained with DAPI to label DNA and an automated cell counting workflow was developed using the open-source software FIJI and TrackMate plugin to count individual cells based on fluorescent intensity and custom python scripts were used for analysis (**Supplemental Material 3.S1 and 3.S2**) [15]. *euo*prom-mNG(LVA)+/ DAPI+ cells were counted as RBs and *hctA*prom-mNG(LVA)+/DAPI+ cells as IBs. RB and IB numbers were quantified on a per-inclusion basis. As suggested by the live-cell kinetic data, the RB population first amplified until ~26 hpi at which point RB numbers plateaued (**Fig. 3.2A**). There was, however, large variation between inclusions, yet the average number of RBs present throughout the plateau period (>28 hpi) was relatively stable at ~30 individuals. Our previously published live-cell data indicated that this variation in *euo*prom expression occurs on a per-inclusion level, with RB-associated fluorescent expression being static per-inclusion but varying between inclusions [9]. Consistent with the live-cell *hctA*prom-mNG(LVA) kinetics, the IB population also increased in number until ~26 hpi and then plateaued (**Fig. 3.2B**). The number of IBs during the plateau period also exhibited large variation per inclusion, with an average number of IBs at ~20 individuals.

**Modeling single-cell chlamydial development.** To explore the relationship between cell division and RB-to-IB development, we created two agent-based models (ABMs) to simulate the chlamydial developmental cycle. Model construction was performed using the Python-based platform Cellmodeller and analyzed with custom python scripts (**Supplemental Material 3.S3**) [16]. Individual RBs (RB$_R$s and RB$_E$s), IBs, and EBs were simulated as well as their intracellular Euo, HctA, and HctB protein concentrations. Multiple aspects of these models were informed by confocal and kinetic data using our three dual reporter strains. The average RB number during the plateau period from our fixed-cell *euo*prom-mNG(LVA) promoter-

reporter experiment (**Fig. 3.2A**) was used for the average number of total simulated RBs, and the live-cell *euo*prom-mNG(LVA) expression data was fit to a mathematical function and used to drive the $RB_R$-to-$RB_E$ maturation process (**Supplemental Material 3.S4**). We simulated IB-to-EB development using a direct maturation mechanism, as we previously showed that IB-to-EB development takes ~8 h to occur after cell division inhibition [9]

As the mechanism of RB-to-IB development is currently unknown, we designed our models to simulate two opposing mechanisms of IB development: asymmetric IB production from RBs and direct conversion of RBs to IBs. In the asymmetric production model, IBs are produced by the subset of mature reticulate bodies ($RB_E$) via asymmetric/polarized cell division. Where, upon each division event one $RB_E$ and one IB daughter cell are produced (**Fig. 3.3A**)(**Supplemental Material 3.S5**). Whereas, in the direct conversion model, independent RBs 'decide' to convert directly into IBs (**Fig. 3.3B**), using a coin flip decision mechanism (**Supplemental Material 3.S6**). Both models were capable of emulating the live-cell promoter-reporter data (**Fig. 3.3CD**), however the direct conversion model had to be highly constrained. For the direct conversion model we had to match the decision time of RB-to-IB conversion to the time to cell division. When we set the decision time to half of that of cell division, the RB numbers drove to extinction as conversion outcompeted replication (**Fig. 3.S1A**). Conversely, when we set decision time to twice as long as the division time, the RB, IB, and EB cell numbers continually increased throughout simulated development (**Fig. 3.S1B**).

The asymmetric production and the direct conversion models were capable of emulating the live-cell promoter-reporter data, giving us an *in silico* platform to test the developmental hypotheses at a single-cell level.

**Simulated inclusion-level and single-cell kinetics.** Although the asymmetric production and direct conversion models produced similar kinetic results on average, there were noticeable differences between the simulated cell-form subpopulation outputs of individual inclusions (**Fig. 3.4**). The asymmetric production model predicted stable RB numbers after total maturation to the $RB_E$ (**Fig. 3.4A**). Where,

the direct conversion model exhibited large fluctuations in RB numbers throughout the entirety of the simulation (**Fig. 3.4C**). Simulated individual inclusions also differed in their EB kinetic outputs. In the asymmetric production model, the EB production slopes were linear (**Fig. 3.4B**); while, in the direct conversion model, the individual inclusion traces were not (**Fig. 3.4D**).

Further differences were seen at the single-cell level, where, in the asymmetric production model, individual RBs were a stem cell-like population that produced an IB upon cell division event (**Mov. 3.S1**). This differed from the direct conversion model, where individual RBs were transient as they converted to IBs (**Mov. 3.S2**).

The differences between the simulated outputs from each model suggested that the mechanism of RB-to-IB development could be determined experimentally by observing the cell-form specific dynamics at the individual inclusion and single-cell level.

**Individual inclusion RB populations are stable and EB production is linear.** The two ABMs predicted vastly different RB and EB subpopulation kinetics at the individual inclusion level. To determine which hypothesis was more likely, expression of the *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 (RB/EB) promoter-reporter strain was monitored within individual inclusions via live-cell fluorescence microscopy from 10-50 hpi (**Fig. 3.5**). Fluorescence intensity from *euo*prom-mNG(LVA) began between ~10-18 hpi and plateaued at ~24-30 hpi, dependent on the inclusion (**Fig. 3.5A**). After *euo*prom-mNG(LVA) expression plateaued, individual inclusion traces demonstrated little variation in fluorescence intensity levels. The minimal wobble seen in *euo*prom-mNG(LVA) fluorescence intensity was due to the small variation z-slice focus. Initiation of fluorescent expression of individuals from *hctB*prom-mKate2 was detected between ~24-28 hpi and continued linearly throughout the remainder of development (**Fig. 3.5B**). The lack of variation in individual inclusions from the *euo*prom-mNG(LVA) promoter-reporter after plateau and the linear production of *hctB*prom-mKate2 fluorescence was in agreement with the asymmetric production model.

**The individual RBs are static and IBs are transient.** Simulations of the asymmetric production and direct conversion ABMs also predicted differing cell-form fates at a single-cell level. The dynamics of individual cells could not be followed with fixed confocal imaging due to the destructive nature of this method. Therefore, to test the model predictions experimentally, automated fluorescence live-cell microscopy was used to follow individual chlamydial cells within inclusions during active infections. The *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2: RB/EB, or *Ctr*-L2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2: IB/EB, dual promoter-reporter strains were used to infect host cells, and images were taken in 15 min intervals starting at 24 hpi and continuing until 60 hpi. These images produced timelapse videos of single RBs and IBs within individual inclusions (**Mov. 3.S3 and 3.S4**). Representative images were selected from this timelapse experiment at 5 h intervals to demonstrate the individual cell-form dynamics (**Fig. 3.6**). Individual RBs were easily tracked from one frame to the next and remained static throughout the infectious cycle (**Fig. 3.6A, Mov. 3.S3**). This differed from the individual IB dynamics, where individuals expressing *hctA*prom-mNG(LVA) were transient, appearing and disappearing sporadically within inclusions through time (**Fig. 3.6B, Mov. 3.S4**). Both promoter-reporter strains increased in *hctB*prom-mKate2 expression levels throughout the infections, indicating that the developmental cycle was able to continue unhampered (**Fig. 3.6AB: second row**). The stability of individual RBs was again in agreement with the predicted outcomes from the asymmetric production model. The transience of *hctA*prom-mNG(LVA)+ cells (IBs) as well as the increase in *hctB*prom-mKate2 (EB) expression suggested that the IB is a transient cell form and that IB-to-EB development may be occurring through direct maturation.

**Simulating inhibition of cell division.** The fundamental difference between the direct conversion and asymmetric production hypotheses is their dependency on cell division. In the direct conversion hypothesis, RBs transition directly into IBs independent of cell division. Conversely, in the asymmetric production hypothesis cell division must occur for IBs to be produced. To determine whether cell division was needed in the development of IBs, or if RBs were capable of directly converting

into IBs, inhibition of RB cell division was simulated in each model at 30 hpi (**Fig. 3.7**) (**Supplemental Material 3.S5 and 3.S6**). The asymmetric production model predicted that RB numbers would be unchanged after cell division inhibition, but an immediate drop in IB numbers would occur as the $RB_E$s could no longer produce IBs, yet the IBs would still convert into EBs. The EB population was predicted to continue linearly, identical to the untreated simulation until the IB population went extinct, at which point EB numbers would abruptly plateau (**Fig. 3.7A**). The direct conversion model predicted that once cell division was inhibited RBs would decrease in number as they continued to convert into IBs, eventually reaching extinction ~10 h post cell division inhibition. The direct conversion of RBs into IBs created a delay in the IB extinction event and produced a logistic slope for EB production (**Fig. 3.7B**). The simulation of cell division inhibition in each model produced separate predicted developmental outcomes, allowing for experimental differentiation of the asymmetric production and direct conversion hypotheses.

**RBs do not convert into IBs after cell division inhibition.** Each model predicted differing RB population dynamics after cell division inhibition. The asymmetric production model predicted unchanged RB numbers after treatment, whereas the direct conversion model predicted that RB numbers would decrease over time as RBs continued to convert into IBs. Therefore, to test these predictions experimentally, two known cell replication inhibitors were used: penicillin and ciprofloxacin. *Chlamydia* is a unique bacteria in that it does not contain a peptidoglycan cell wall and instead uses peptidoglycan only in septum formation [17]. Previously published data has shown that *Chlamydia* treated with penicillin is inhibited in division. However, although unable to divide, these *Chlamydia* still increase in biomass, replicate their DNA, and produce RB-like gene expression profiles [9, 18, 19]. Ciprofloxacin has been shown to prevent bacterial DNA replication by inhibiting topoisomerase and DNA-gyrase [20]. Mirroring the simulated experiments, 60X live-cell images were taken of *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 infected host cells treated with either penicillin or ciprofloxacin at two timepoints: at antibiotic treatment (30 hpi) and 10 h later (40 hpi) (**Fig. 3.8**). The number of

*euo*prom-mNG(LVA)+ cells (RBs) was quantified on a per-inclusion basis. Consistent with the confocal time-series experiment (**Fig. 3.2**), there was large variation in RBs numbers between individual inclusion, ranging from a single RB to greater than 50 RBs (**Fig. 3.8A**). However the number of RBs per inclusion remained consistent across timepoints (**Fig. 3.8A**). Neither the vehicle-only or penicillin-treated samples exhibited a decrease or increase in RB numbers, with the ratio of RBs per time point being ~1:1 (**Fig. 3.8B**). Surprisingly, after treatment with ciprofloxacin the number of *euo*prom-mNG(LVA)+ cells approximately doubled (**Fig. 3.8AB**). To confirm that chlamydial DNA replication was inhibited by ciprofloxacin, digital droplet (ddPCR) was performed on *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 infected samples treated with either ciprofloxacin, penicillin, or mock at 30 hpi. Host monolayers were harvested every 4 h from 26-54 hpi. As previously reported, genome copy number continued to increase in the penicillin-treated samples [9, 18, 19]. There was however a large reduction in genome copies after ciprofloxacin treatment when compared to the mock and penicillin-treated samples (**Fig. 3.S2**). The stability of intra-inclusion RB numbers on a per-inclusion basis and lack of depletion in RBs after cell division inhibition strongly suggests that RBs are not directly converting into IBs and that cell division is required for production of the IB.

**Development of the IB requires cell division.** After simulated cell division inhibition both models predicted a decrease in IB numbers as they converted into the EBs (**Fig. 3.7**), making it difficult to discern which mechanism is utilized in development of the IB. However, along with total IB numbers, the IB (*hctA*) promoter-reporter was also simulated. Although the direct conversion model had predicted a depletion in IB numbers after cell division inhibition, total accumulation of the simulated *hctA*prom-GFP was predicted to continue to increase as RBs continued converted to IBs post division inhibition. This increase in total simulated *hctA*prom-GFP eventually plateaued ~12 h later, after the entire RB population went extinct (**Fig. 3.9A**). This differed from the asymmetric production model, where an almost immediate halt in simulated *hctA*prom-GFP accumulation was predicted after division inhibition, as IBs could no longer be produced (**Fig. 3.9B**). As a control, we

also simulated RB cell death in both models. Elimination of the RB population predicted that further development of the IB would be prevented regardless of the underlying developmental mechanism, resulting in a predicted *hctA*prom-GFP output identical to that of the asymmetric production hypothesis (**Fig. 3.9, RB Death: orange**). The effects of simulating cell division inhibition or RB death were mirrored in the predicted EB cell number outputs (**Fig. 3.9**).

To assay for the effects of RB death, we developed an inducible system to kill the dividing chlamydial population. Penicillin-binding proteins (PBPs) are a suite of enzymes involved in peptidoglycan synthesis and cell division [21]. When overexpressed, PBP3 has been shown to induce lysis in dividing cells [22]. The ORF of PBP3, *ftsI*, was tagged with a C-terminal 3XFLAG epitope and placed under translational control using our previously published E-riboswitch chlamydial system [13]. The *euo*prom-mNG(LVA)_*hctB*prom-mKate2 promoter-reporter cassette was cloned into the E-riboswitch-*ftsI*3XFLAG vector and transformed into *Chlamydia trachomatis* L2. Lysis of the RB population by induction of FtsI3XFLAG was confirmed by immunofluorescence confocal imaging of host cells infected with *Ctr*-L2-E-*ftsI*3XFLAG_*euo*prom-mNG(LVA)_*hctB*prom-mKate2 (**Fig. 3.S3**). Infected cells were treated with vehicle-only or induced for FtsI3XFLAG at 20 hpi. To visualize the presence and absence of FtsI induction as well as the total *Chlamydia* present, samples were labeled with a monoclonal anti-FLAG antibody and stained with DAPI. The 20 hpi samples contained multiple *euo*prom-mNG(LVA)+ cells per inclusion, indicating active transcription and translation. There was also no 3XFLAG detection in the 20 hpi samples (**Fig. 3.S3A**). The uninduced 30 hpi samples exhibited a similar phenotype to the 20 hpi samples: *euo*prom-mNG(LVA)+ cells, and no FLAG detection (**Fig. 3.S3B: uninduced**). Conversely, inclusions from the 30 hpi FtsI induced samples were relatively empty compared to their 30 hpi uninduced counterparts and 3XFLAG detection was present in a subset of cells, indicating FtsI overexpression. The cells that stained positively for 3XFLAG were RB-like in size, however these cells were misshapen and contained little to no fluorescence from the *euo*prom-mNG(LVA) reporter, indicating a lack of active expression and suggesting chlamydial lysis had occurred (**Fig. 3.S3B: induced**). Digital droplet PCR was used

to confirm that over-expression of FtsI inhibited chlamydial DNA replication (**Fig. 3.S4**).

To experimentally test the effects of cell division inhibition and RB lysis on IB formation, a photostable IB promoter-reporter variant, *hctA*prom-mEos3.2, was used to replace *euo*prom-mNG(LVA) in the E-*ftsI*3XFLAG_*euo*prom-mNG(LVA)_*hctB*prom-mKate2 construct. Mirroring the simulated experiments, *Ctr*-L2-E-*ftsI*3XFLAG_*hctA*prom-mEos3.2_*hctB*prom-mKate2 was used to infect host cells and treated with penicillin, ciprofloxacin, or induced for FtsI expression at 30 hpi. Fluorescence of the *hctA*prom-mEos3.2 was quantified using automated live-cell microscopy. Expression from *hctA*prom-mEos3.2 halted immediately after exposure to penicillin, ciprofloxacin and FtsI induction (**Fig. 3.10A**). However, as previously reported, reinitiation of expression from the *hctA* promoter occurred in the large aberrant RBs ~10 h post penicillin treatment (**Mov. 3.S5**) [9]. EB production was measured by infectious forming units (IFU) assay. Monolayers were harvested every 4 hours from 10 to 50 hpi and used to infect fresh host cells for EB quantification. Regardless of treatment, EB development continued for ~8 hours and then plateaued (**Fig. 3.10B**). These results were in agreement with the asymmetric production model and indicate that cell division is required for IB production. The continued increase and plateau EB numbers ~8 hours post all treatments also suggests that IBs may be maturing directly into EBs.

**IB-to-EB development is cell division independent.** Results from the live-cell *hctA*prom-mEos3.2 and IFU experiments suggested that although production of further IBs requires cell division, IB-to-EB development is occurring via direct maturation, and thus a cell division independent process. To determine whether IB-to-EB development was occurring post cell division, *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 was used to infect host cells and treated with either ciprofloxacin or penicillin at 20 hpi. Previously published data and our current results suggested that IB-to-EB maturation takes ~8-10 h post treatment[9]. Therefore, samples were fixed for confocal imaging and stained for DNA (DAPI) at 20 hpi (at treatment) and 30 hpi (**Fig. 3.11**). The presence of *euo*prom-mNG(LVA)+ cells were

used to indicate RBs and *hctB*prom-mKate2+ cells for EB development. In the 20 hpi samples, there were ~20 *euo*prom-mNG(LVA)+ cells (RBs) in each inclusion. These inclusions also contained a large proportion of smaller chlamydial cells which exhibited only DAPI staining (**Fig. 3.11A**). At 30 hpi, the mock-treated samples contained *euo*prom-mNG(LVA)+, DAPI+ only, and *hctB*prom-mKate2+ cells (**Fig. 3.11B**). Total inclusion size in the ciprofloxacin and penicillin-treated samples was approximately equal to the mock-treated samples, however the inclusions from the ciprofloxacin and penicillin-treated samples were relatively empty, indicating cell division inhibition had occurred. As previously reported, the RBs in the penicillin-treated samples had developed into large aberrant cells (**Fig. 3.11B:PEN**) [9, 18]. Surprisingly, although chlamydial replication had been inhibited, the RBs in the ciprofloxacin-treated samples more closely resembled the mock-treated RBs in size (**Fig. 3.11B:CIP**). Compared to the 20 hpi samples, both the 30 hpi ciprofloxacin and penicillin samples demonstrated a substantial decrease in the number of DAPI only cells and an increase in *hctB*prom-mKate2+ cells, suggesting that these smaller DAPI only cells were possibly IBs and had developed into EBs post cell division inhibition.

To confirm that EB development was in fact occurring post cell division, RB cell death was induced utilizing our E-*ftsI*3XFLAG overexpression system. Host cells were infected with *Ctr*-L2-E-*ftsI*3XFLAG_*euo*prom-mNG(LVA)_*hctB*prom-mKate2. Fixed confocal images were taken at induction (20 hpi) and 10 h post induction (30 hpi)(**Fig. 3.12**). Infected cells were stained for DNA and 3XFLAG detection. The 20 hpi sample mirrored that of the 20 hpi sample from the ciprofloxacin and penicillin-treated experiment (**Fig. 3.11A**), where inclusions contained *euo*prom-mNG(LVA)+ cells and DAPI only cells (**Fig. 3.12A**). The 30 hpi uninduced sample also appeared wildtype in cell form and inclusion presentations: anti-FLAG negative, *euo*prom-mNG(LVA)+ cells, *hctB*prom-mKate2+ cells, and DAPI only cells (**Fig. 3.12B:uninduced**). The 30 hpi FtsI induced sample contained positive anti-FLAG labeling in a subset of cells, indicating FtsI over-expression. As in the 30 hpi ciprofloxacin and penicillin treatments, the 30 hpi FtsI induced sample exhibited a decrease in DAPI only cells, yet an increase in *hctB*prom-mKate2+ cells (**Fig.**

**3.12B:induced**). This increase in the number of *hctB*prom-mKate2+ cells from DAPI only cells post RB lysis again suggests that IB-to-EB development is occurring after cell division and is independent of this process.

**IB-to-EB development occurs by direct maturation**. To further determine whether IBs were maturing directly into EBs independent of cell division, colocalization of the photostable IB (*hctA*prom-mEos3.2) and EB (*hctB*prom-mKate2) reporters was evaluated. *Ctr*-L2-E-*ftsI*3XFLAG_*hctA*prom-mEos3.2_*hctB*prom-mKate2 was used to infect Cos-7 cells. Due to *hctA* expression in penicillin-treated aberrant RBs, cell division was inhibited with only ciprofloxacin treatment or induced for FtsI over-expression. Infected cells were treated at 18 hpi (corresponding to initial IB production), fixed for confocal imaging at 22 and 34 hpi, and counterstained for DNA and 3XFLAG expression. At 22 hpi (4 h post treatment) all treatments consisted of inclusions that contained *hctA*prom-mEos3.2+ cells (IBs) and a subset of DAPI-only large RB-like cells. EBs (*hctB*prom-mKate2+ cells) were absent at this time point, consistent with previous data showing *hctB* expression initiates at ~24 hpi (**Fig. 3.13A:UNT, CIP**) [9]. The RB-like cells in the FtsI induced sample also exhibited positive anti-FLAG staining (**Fig. 3.13A:FstI**). In the 34 hpi ciprofloxacin-treated and FtsI induced samples inclusions appeared relatively empty compared to the mock-treated sample (**Fig. 3.13B**). Expression from *hctA*prom-mEos3.2 and *hctB*prom-mKate2 colocalized within individual cells. There was, however, variation in individual cell size and the intensities of *hctA*prom-mEos3.2 and *hctB*prom-mKate2 fluorescence from cell-to-cell, creating a mosaic from yellow-green to dark orange. At 34 hpi, FtsI (anti-FLAG) overexpression appeared in primarily RB-like cells. However, a subset of *hctA*prom-mEos3.2+/*hctB*prom-mKate2+ cells also contained positive anti-FLAG staining, suggesting that, although FtsI over-expression had occurred in the IB, IB-to-EB maturation was unaffected (**Fig. 3.13B: FtsI**). These results support that IB-to-EB development is a cell division independent process and occurs by direct maturation.

**Discussion**

All *Chlamydiae* progress through an intracellular biphasic developmental cycle dependent on two primary cell forms. The EB is responsible for initiating infection of the host. Whereas, the RB must replicate to increase chlamydial numbers. Although this developmental process is essential for chlamydial proliferation and dissemination, chlamydial development is largely understudied and little is known about the mechanisms that regulate it.

Based on data from our previous study, we hypothesized that IB production occurs by asymmetric division from mature RBs, termed $RB_E$s. Multiple studies have demonstrated evidence supporting asymmetric division in *Chlamydia*. Both RBs and EBs have been shown to exhibit polar Type III secretion systems by EM and IFA [23, 24]. Chlamydial cell division has also been shown to be polarized, as the peptidoglycan septum ring is produced at a single pole rather than equatorially, with asymmetric protein expression occurring on either side of the division plane [25, 26].

To further investigate the dynamics of cell-form development in *Chlamydia*, we created multiple cell-form dual reporter chlamydial strains to monitor active cell-form specific expression in real time. Our results showed that active expression was spatially isolated to individual cells with no occurrence of reporter colocalization, indicating the presence of distinct chlamydial subpopulations. Average live-cell inclusion-level kinetics from the *euo*prom-mNG(LVA) reporter was detected beginning at ~11 hpi and reached a maximum at ~24 hpi. After 24 hpi, *euo*prom-mNG(LVA) plateaued until inclusion lysis. Single-cell RB counts throughout the developmental cycle showed that the plateau in *euo*prom-mNG(LVA) corresponded to RB numbers. Live-cell kinetics and single-cell counts from the IB-associated reporter, *hctA*prom-mNG(LVA), also plateaued late in the infectious cycle (>30 hpi). Both RB and IB counts demonstrated large variability in cell numbers between inclusions per time point, suggesting that $RB_R$-to-$RB_E$ maturation rates may fluctuate on a per-inclusions basis.

To determine the role of cell division in RB-to-IB development, we created two agent-based models that simulated RB-to-IB differentiation using either a direct conversion or asymmetric division mechanism. Live-cell data and single-cell counts

from each reporter strain were used to inform the cell-form kinetics for each model. Both models produced near identical aggregate outputs and emulated the cell-form kinetics of the developmental cycle. However, the direct conversion model was only capable of these results when the RB-to-IB conversion decision time matched the time to RB cell division. This constraint on the direct conversion model suggested that development of the IB may be a cell division dependent process.

The asymmetric production and direct conversion models produced dynamic and behavioral differences at the subpopulation and individual level, suggesting we could observationally distinguish between the two mechanisms of development. Individual inclusion traces from the asymmetric production model indicated that the RB population would reach a maximum number and then remain static throughout the remainder of the infection as $RB_E$s divided asymmetrically to produce IBs. This asymmetric production of IBs also led to a linear increase in EBs within individual simulated inclusions. Conversely, individual traces from the direct conversion model indicated that the RB population would continually fluctuate throughout the infectious cycle as the rate of RB replication matched IB conversion. The balancing of RB replication with IB conversion produced runs of RBs and IBs throughout the infectious cycle and led to non-linear EB production on an individual inclusion level. Analysis of single-inclusion traces from *euo*prom-mNG(LVA) showed that each RB population reached a maximum between ~24-30 hpi and remained static thereafter. Our single-cell data also indicated that individual RBs were static late in the infectious cycle (>24 hpi). Inclusion-level analysis of *hctB*prom-mKate2 demonstrated a linear increase in expression on a per inclusion basis. These observational results suggest that asymmetric division is the likely mechanism of IB development.

The direct conversion and asymmetric division models differed only in their dependence on cell division for IB development. Therefore, we determined that we could differentiate between the two mechanisms by simulating cell division inhibition. To test this experimentally, penicillin and ciprofloxacin were used to target different chlamydial replication pathways (penicillin: cell division and ciprofloxacin:vDNA replication, respectively). In both the untreated and penicillin-treated samples, the

number of RBs did not decrease per inclusion. Surprisingly, in the ciprofloxacin treated samples, the number of RB-like cells consistently doubled. IBs are known to undergo large morphological changes, including DNA condensation [4, 7]. DNA gyrase has been shown to aid in DNA condensation; and therefore, treatment with ciprofloxacin may be preventing proper DNA condensation, leading to a halt in the development of newly produced IBs [27]. Our data also showed that development of the IB was dependent on cell division, for once cell division was inhibited, or the RB population was killed (FtsI induction), further IB production, as measured by *hctA*prom-mEos3.2 accumulation, immediately halted. The culmination of these results strongly suggests that RBs are not converting directly to IBs and that asymmetric division is the likely mechanism for development of IBs.

We previously showed that EB development continues for ~8 h post treatment with penicillin[9]. We hypothesized that this phenomenon was due to EB development being a committed step that occurs post cell division by direct maturation from the IB. IFU results from this study exhibited similar trends, with cell division inhibited or RB-lysed (FtsI induced) *Chlamydia* demonstrating an increase in infectious progeny for ~8 h post treatment. Single-cell analysis revealed that EB development, as demonstrated by the accumulation of small DNA dense *hctB*prom-mKate2 expressing cells, was also occurring post cell division inhibition and after RB lysis. We further showed that, after cell division inhibition and RB death, colocalization of the IB and EB reporters (*hctA*prom-mEos3.2 and *hctB*prom-mKate2, respectively) occurred. These data provide direct evidence that IB-to-EB development is a cell division independent process and that IBs mature directly into EBs. Live-cell analysis of the active IB reporter, hctAprom-mNG(LVA), also showed that individual IBs demonstrate transient *hctA* promoter expression throughout the developmental cycle, suggesting that after IB production, *hctA* expression is repressed as IBs continue their development into EBs. Soules et al. presented further evidence of the involvement of a regulatory cascade in IB-to-EB maturation, where they showed that *Chlamydia's* only two component regulatory system, CtcB/ctcC is expressed at a time concurrent with IB formation (>18 hpi) and that CtcC, the response regulator, is

upstream of $\sigma^{54}$ and positively regulates a large subset of $\sigma^{54}$-dependent EB-associated genes [28, 29]

Although the mechanisms that control asymmetric division in *Chlamydia* are unknown, *Chlamydia* does encode many regulatory pathways that are homologous to those found in the asymmetrically dividing model bacteria, *Bacillus subtilis* and *Caulobacter crescentus*. Of these, are two alternative sigma factors, $\sigma^{28}$ and $\sigma^{54}$, which have been shown to control the expression of late EB-associated genes [28, 30, 31]. In *B. subtilis,* $\sigma^{28}$ is involved in the early stage of asymmetric division and responsible for initiation of sporulation [32, 33, 34]. Whereas, in *C. crescentus*, ChpT/CtrA-$\sigma^{N}$, which are homologous to CtcB/CtcC-$\sigma^{54}$, are involved in cell cycle and cell-form specific gene regulation of the flagellum [35]. Both regulatory control systems of $\sigma^{28}$ and $\sigma^{54}$ in *B. subtilis* and *C. crescentus* have been thoroughly described and may provide crucial insights into the upstream drivers of $\sigma^{28}$ and $\sigma^{54}$ in *Chlamydia*. *B. subtilis* and *C. crescentus* also utilize an array of other processes to control cell-form development including methylation dependent transcription, localized protein degradation and protein sequestration [35, 36]. Given that *Chlamydia* appears to undergo asymmetric division to produce the IB cell form and that *Chlamydia* contains several components that are utilized by other asymmetrically dividing bacteria, the regulatory mechanisms that these model organisms employ should be used as a guide in the future research of chlamydial cell-form development.

## Materials and Methods

**Organisms and cell culture.** Cos-7 cells were obtained from (ATCC). Cells were maintained in a 5% CO2 incubator at 37°C in RPMI 1640 (Cellgro) supplemented with 10% fetal plex (FP) and 10g/ml gentamicin. All *C. trachomatis*-L2 (LGV Bu434) strains were grown in and harvested from Cos-7 cells. Elementary bodies were purified by density centrifugation using 30% MD-76R at 48 hours post infection [10]. Purified elementary bodies were stored at -80°C in sucrose-phosphate-glutamate buffer  (10 mM sodium phosphate [8 mM K 2HPO 4, 2 mM KH 2PO 4], 220 mM sucrose, 0.50 mM L-glutamic acid; pH 7.4). *Escherichia coli* ER2925 (*dam-/dcm-*) was utilized to produce unmethylated constructs for transformation into *Chlamydia*.

**Promoter-reporter and inducible expression constructs.** All constructs were created using the p2TK2SW2 plasmid [37]. All promoters and the *ftsI* ORF were originally amplified from *C. trachomatis*-L2 (LGV Bu434) genomic DNA using the indicated primers (**Table ST 3.1**). Fluorescent reporters were ordered as gBlocks and cloning was performed with the In-fusion HD EcoDry Cloning kit (Takara). Promoter-reporter constructs were created as previously described [9, 38]. The original p2TK2SW2-E-*ftsI*3XFLAG was generated by inserting the *ftsI* ORF into the previously created p2TK2SW2-E-clover-3XFLAG plasmid between the tn5 promoter/ E-riboswitch and 3XFLAG [13]. Following this, was the insertion of each dual promoter-reporter (*euo*prom-mNG(LVA)_*hctB*prom-mKate2 and *hctA*prom-mEos3.2_*hctB*prom-mKate2) upstream of E-*ftsI*3XFLAG and in reverse orientation.

**Chlamydial transformation and isolation.** Transformation of *C. trachomatis*-L2 was performed as previously described with selection using 500 ng/ul spectinomycin [37]. Clonal isolation of transformants was achieved by inclusion isolation (MOI <1) via micro-manipulation. To confirm conality each construct was purified from the chlamydial transformants, transformed into *E. coli* and five colonies were sequenced.

**Infections.** Infections were synchronized by incubating Cos-7 cells with *C. trachomatis*-L2 EBs in Hank's balanced salt solution (HBSS) (Gibco) for 15 minutes at 37ºC while rocking. The inoculum was removed and cells were washed with prewarmed (37ºC) HBSS. The HBSS was replaced with fresh RPMI-1640 containing

10% FP, 10 µg/ml gentamicin, 1 µg/ml cycloheximide, and 1 mg/ml heparin sodium. For cell division experiments chlamydial cell division was inhibited by the addition of 0.5 µg/ml ciprofloxacin or 1 U/ml penicillin-G to the media. Expression of *ftsI*/3XFLAG was induced by the addition of 0.5 mM theophylline to the media [13].

**Replating assays.** Strain specific EBs were isolated from infected Cos-7 cells by scraping the host monolayer and pelleting via centrifugation at 4ºC for 30 min at 18213 rcfs. EB pellets were resuspended in 4ºC RPMI via sonication and used to infect Cos-7 cells in polystyrene 96-well microplates in a 2-fold dilution series. Infected plates were incubated for 29 hours following fixation in methanol. Fixed cells were stained with 4′,6-diamidino-2-phenylindole (DAPI) for visualization of host-cell nuclei and anti-MOMP antibody conjugated to FITC (Thermo Scientific™) for visualization of EBs and inclusion counts. Monolayers were imaged with an Andor Zyla sCMOS and Nikon Eclipse TE300 inverted microscope utilizing a scopeLED lamp at 470nm and 390nm, and BrightLine band pass emissions filters at 514/30nm and 434/17nm. Automated image acquisition was performed using µManager software [39]. Inclusion numbers were quantified with custom scripts in ImageJ and analyzed in custom Python notebooks as previously described [9, 38, 40]

**Genome number quantification.** Total DNA was isolated from infected Cos-7 cells during active infections using an Invitrogen Purelink genomic DNA mini kit. A QX200 digital droplet system (BioRad) was utilized for quantification of chlamydial genomic copies. A 2X ddPCR™ Supermix for Probes-No dUTP kit (BioRad) and a custom *copN*-specific primer/probe set was used for DNA detection (**Table ST 3.1**).

**Live-cell microscopy.** Monolayers were infected with synchronized *Ctr*-L2 EBs and grown on multi-well glass-bottom plates. Infections were grown in an OKOtouch CO2/heated stage incubator. Fluorescence images were acquired via epifluorescent microscopy using a Nikon Eclipse TE300 inverted microscope with a ScopeLED lamp at 470nm and 595nm, and BrightLine Bandpass filters at 514/30nm and 590/20nm. 20X/0.4NA dry, 40X/0.6NA dry, and 60X/1.40NA oil objective lenses were used. DIC was used to auto-focus images. Image acquisition was performed using an Andor Zyla sCMOS camera in conjugation with µManager software [39]. Images were taken in 30 min intervals, unless otherwise stated. Imaging ranged from 10 to

60 hours after *Ctr*-L2 infection, depending on the experiment. Multiple fields were imaged for each treatment and the fluorescent intensity of individual inclusions was monitored using the Trackmate plug-in in ImageJ [15]. Inclusion fluorescent intensities were averaged and graphed in Python as previously described [38]

**Confocal microscopy.** Cos-7 cells were seeding on to glass coverslips and infected with *Ctr*-L2. Samples were fixed at the designated times in 2% paraformaldehyde in filtered phosphate-buffered saline (PBS) at room temperature, overnight. Samples were then washed with filtered PBS and stained with DAPI to visualize DNA and monoclonal anti-FLAG M2 antibody (Sigma, Thermo Scientific™) with alexa 647 anti-mouse secondary antibody to visualize FtsI3XFLAG expression. Coverslips were mounted onto a microscope slide using MOWIOL (100 mg/mL 150 MOWIOL® 4-88, 25% glycerol, 0.1 M Tris pH 8.5). Images were acquired using a Nikon spinning disk confocal inverted microscope with a 100X oil objective with a laser lamp at 405nm, 490nm, 568nm and 660nm. Image acquisition was performed using an Andor Ixon EMCCD camera and the Nikon elements software. Multiple inclusions were imaged for each treatment/time point and quantification of individual cells was performed using Trackmate. Chlamydial cell numbers were then analyzed in custom Python notebooks. Representative confocal micrographs are maximal intensity projections of 3D data sets.

**Data Availability.** All data, bacterial strains and methodologies are available upon request.

## Acknowledgments

**References**

1. Global prevalence and incidence of selected curable sexually transmitted diseases: overview and estimates. Published April 29, 2022.

2. National Academies of Sciences E, Medicine;Health, on Population Health MD, on Prevention PHP, in the United States C of STI, Crowley JS, Geller AB, Vermund SH. Sexually Transmitted Infections: Adopting a Sexual Health Paradigm. Published online March 24, 2021.

3. Clifton DR, Fields KA, Grieshaber SS, Dooley CA, Fischer ER, Mead DJ, Carabeo RA, Hackstadt T. A chlamydial type III translocated protein is tyrosine-phosphorylated at the site of entry and associated with recruitment of actin. *Proceedings of the National Academy of Sciences of the United States of America*. 2004;101(27):10166-10171. doi:10.1073/pnas.0402829101

4. Friis RR. Interaction of L cells and Chlamydia psittaci: entry of the parasite and host responses to its development. *Journal of bacteriology*. 1972;110(2). doi:10.1128/jb.110.2.706-721.1972

5. Grieshaber NA, Runac J, Turner S, Dean M, Appa C, Omsland A, Grieshaber SS. The sRNA Regulated Protein DdbA Is Involved in Development and Maintenance of the Chlamydia trachomatis EB Cell Form. *Frontiers in Cellular and Infection Microbiology*. 2019;11. doi:10.3389/fcimb.2021.692224

6. Belland RJ, Zhong G, Crane DD, Hogan D, Sturdevant D, Sharma J, Beatty WL, Caldwell HD. Genomic transcriptional profiling of the developmental cycle of Chlamydia trachomatis. *Proceedings of the National Academy of Sciences of the United States of America*. 2003;100(14):8478-8483. doi:10.1073/pnas.1331135100

7. Lee JK, Enciso GA, Boassa D, Chander CN, Lou TH, Pairawan SS, Guo MC, Wan FYM, Ellisman MH, Stterlin C, Tan M. Replication-dependent size reduction precedes differentiation in Chlamydia trachomatis. *Nature communications*. 2018;9(1). doi:10.1038/s41467-017-02432-0

8. Shaw EI, Dooley CA, Fischer ER, Scidmore MA, Fields KA, Hackstadt T. Three temporal classes of gene expression during the Chlamydia trachomatis developmental cycle. *Molecular microbiology*. 2000;37(4):913-925. doi:https://

doi.org/10.1046/j.1365-2958.2000.02057.x

9. Chiarelli TJ, Grieshaber NA, Omsland A, Remien CH, Grieshaber SS. Single-Inclusion Kinetics of Chlamydia trachomatis Development. *mSystems*. 2020;5(5). doi:10.1128/mSystems.00689-20

10. Grieshaber S, Grieshaber N, Yang H, Baxter B, Hackstadt T, Omsland A. Impact of Active Metabolism on Chlamydia trachomatis Elementary Body Transcript Profile and Infectivity. *Journal of bacteriology*. 2018;200(14). doi:10.1128/JB.00065-18

11. Rosario CJ, Tan M. The early gene product EUO is a transcriptional repressor that selectively regulates promoters of Chlamydia late genes. *Molecular microbiology*. 2012;84(6):1097-1107. doi:10.1111/j.1365-2958.2012.08077.x

12. Shaner NC, Lambert GG, Chammas A, Ni Y, Cranfill PJ, Baird MA, Sell BR, Allen JR, Day RN, Israelsson M, Davidson MW, Wang J. A bright monomeric green fluorescent protein derived from Branchiostoma lanceolatum. *Nature methods*. 2013;10(5). doi:10.1038/nmeth.2413

13. Grieshaber NA, Chiarelli TJ, Appa CR, Neiswanger G, Peretti K, Grieshaber SS. Translational gene expression control in Chlamydia trachomatis. *PloS one*. 2022;17(1). doi:10.1371/journal.pone.0257259

14. Shcherbo D, Murphy CS, Ermakova GV, Solovieva EA, Chepurnykh TV, Shcheglov AS, Verkhusha VV, Pletnev VZ, Hazelwood KL, Roche PM, Lukyanov S, Zaraisky AG, Davidson MW, Chudakov DM. Far-red fluorescent tags for protein imaging in living tissues. *The Biochemical journal*. 2009;418(3). doi:10.1042/BJ20081949

15. Tinevez JY, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, Bednarek SY, Shorte SL, Eliceiri KW. TrackMate: An open and extensible platform for single-particle tracking. *Methods (San Diego, Calif)*. 2017;115:80-90. doi:10.1016/j.ymeth.2016.09.016

16. Rudge TJ, Steiner PJ, Phillips A, Haseloff J. Computational modeling of synthetic microbial biofilms. *ACS synthetic biology*. 2012;1(8). doi:10.1021/sb300031n

17. Liechti G, Kuru E, Packiam M, Hsu YP, Tekkam S, Hall E, Rittichier JT,

VanNieuwenhze M, Brun YV, Maurelli AT. Pathogenic Chlamydia Lack a Classical Sacculus but Synthesize a Narrow, Mid-cell Peptidoglycan Ring, Regulated by MreB, for Cell Division. *PLoS pathogens*. 2016;12(5). doi:10.1371/journal.ppat.1005590

18. Skilton RJ, Cutcliffen LT, Barlow D, Wang Y, Salim O, Lambden PR, Clarke IN. Penicillin induced persistence in Chlamydia trachomatis: high quality time lapse video analysis of the developmental cycle. *PLoS ONE*. 2009;4(11). doi:10.1371/JOURNAL.PONE.0007723

19. Ouellette SP, Hatch TP, AbdelRahman YM, Rose LA, Belland RJ, Byrne GI. Global transcriptional upregulation in the absence of increased translation in Chlamydia during IFNgamma-mediated host cell tryptophan starvation. *Molecular microbiology*. 2006;62(5). doi:10.1111/j.1365-2958.2006.05465.x

20. LeBel M. Ciprofloxacin: Chemistry, Mechanism of Action, Resistance, Antimicrobial Spectrum, Pharmacokinetics, Clinical Trials, and Adverse Reactions. *Pharmacotherapy: The Journal of Human Pharmacology and Drug Therapy*. 1987;8(1):3-30. doi:https://doi.org/10.1002/j.1875-9114.1988.tb04058.x

21. Morlot C, Zapun A, Dideberg O, Vernet T. Growth and division of Streptococcus pneumoniae: localization of the high molecular weight penicillin-binding proteins during the cell cycle. *Molecular microbiology*. 2003;50(3). doi:10.1046/j.1365-2958.2003.03767.x

22. Fraipont C, Adam M, Nguyen-Distche M, Keck W, Beeumen JV, Ayala JA, Granier B, Hara H, Ghuysen JM. Engineering and overexpression of periplasmic forms of the penicillin-binding protein 3 of Escherichia coli. *The Biochemical journal*. 1994;298 ( Pt 1)(Pt 1). doi:10.1042/bj2980189

23. Nans A, Saibil HR, Hayward RD. Pathogenhost reorganization during Chlamydia invasion revealed by cryo-electron tomography. *Cellular Microbiology*. 2014;16(10). doi:10.1111/cmi.12310

24. Dumoux M, Nans A, Saibil HR, Hayward RD. Making connections: snapshots of chlamydial type III secretion systems in contact with host membranes. *Current opinion in microbiology*. 2015;23. doi:10.1016/j.mib.2014.09.019

25. Cox JV, Abdelrahman YM, Ouellette SP. Penicillin-binding proteins regulate

multiple steps in the polarized cell division process of Chlamydia. *Scientific reports*. 2020;10(1). doi:10.1038/s41598-020-69397-x

26. Abdelrahman Y, Ouellette SP, Belland RJ, Cox JV. Polarized Cell Division of Chlamydia trachomatis. *PLoS Pathogens*. 2016;12(8). doi:10.1371/journal.ppat.1005822

27. Ohniwa RL, Morikawa K, Kim J, Ohta T, Ishihama A, Wada C, Takeyasu K. Dynamic state of DNA topology is essential for genome condensation in bacteria. *The EMBO Journal*. 2006;25(23). doi:10.1038/sj.emboj.7601414

28. Soules KR, LaBrie SD, May BH, Hefty PS. Sigma 54-Regulated Transcription Is Associated with Membrane Reorganization and Type III Secretion Effectors during Conversion to Infectious Forms of Chlamydia trachomatis. *mBio*. 2020;11(5). doi:10.1128/mBio.01725-20

29. Koo IC, Stephens RS. A developmentally regulated two-component signal transduction system in Chlamydia. *The Journal of biological chemistry*. 2003;278(19):17314-17319. doi:10.1074/jbc.M212170200

30. Yu HHY, Tan M. Sigma28 RNA polymerase regulates hctB, a late developmental gene in Chlamydia. *Molecular microbiology*. 2003;50(2):577-584. doi:10.1046/j.1365-2958.2003.03708.x

31. Yu HHY, Kibler D, Tan M. In silico prediction and functional validation of sigma28-regulated genes in Chlamydia and Escherichia coli. *Journal of bacteriology*. 2006;188(23). doi:10.1128/JB.01082-06

32. Serrano M, Gao J, Bota J, Bate AR, Meisner J, Eichenberger P, Moran CP, Henriques AO. Dual-Specificity Anti-sigma Factor Reinforces Control of Cell-Type Specific Gene Expression in Bacillus subtilis. *PLoS Genetics*. 2015;11(4). doi:10.1371/journal.pgen.1005104

33. Stephens RS, Kalman S, Lammel C, Fan J, Marathe R, Aravind L, Mitchell W, Olinger L, Tatusov RL, Zhao Q, Koonin EV, Davis RW. Genome sequence of an obligate intracellular pathogen of humans: Chlamydia trachomatis. *Science (New York, NY)*. 1998;282(5389). doi:10.1126/science.282.5389.754

34. Kustu S, Santero E, Keener J, Popham D, Weiss D. Expression of sigma 54 (ntrA)-dependent genes is probably united by a common mechanism.

*Microbiological reviews*. 1989;53(3). doi:10.1128/mr.53.3.367-376.1989

35. Biondi EG, Reisinger SJ, Skerker JM, Arif M, Perchuk BS, Ryan KR, Laub MT. Regulation of the bacterial cell cycle by an integrated genetic circuit. *Nature*. 2006;444(7121). doi:10.1038/nature05321

36. Pan Q, Losick R. Unique degradation signal for ClpCP in Bacillus subtilis. *Journal of bacteriology*. 2003;185(17). doi:10.1128/JB.185.17.5275-5278.2003

37. Wang Y, Kahane S, Cutcliffe LT, Skilton RJ, Lambden PR, Clarke IN. Development of a transformation system for Chlamydia trachomatis: restoration of glycogen biosynthesis by acquisition of a plasmid shuttle vector. *PLoS Pathogens*. 2011;7(9). doi:10.1371/journal.ppat.1002258

38. Chiarelli TJ, Grieshaber NA, Grieshaber SS. Live-Cell Forward Genetic Approach to Identify and Isolate Developmental Mutants in Chlamydia trachomatis. *Journal of visualized experiments: JoVE*. 2020;(160). doi:10.3791/61365

39. Edelstein AD, Tsuchida MA, Amodaj N, Pinkard H, Vale RD, Stuurman N. Advanced methods of microscope control using Manager software. *Journal of biological methods*. 2012;1(2).

40. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: an open-source platform for biological-image analysis. *Nature methods*. 2012;9(7). doi:10.1038/nmeth.2019

**Figure 3.1.**
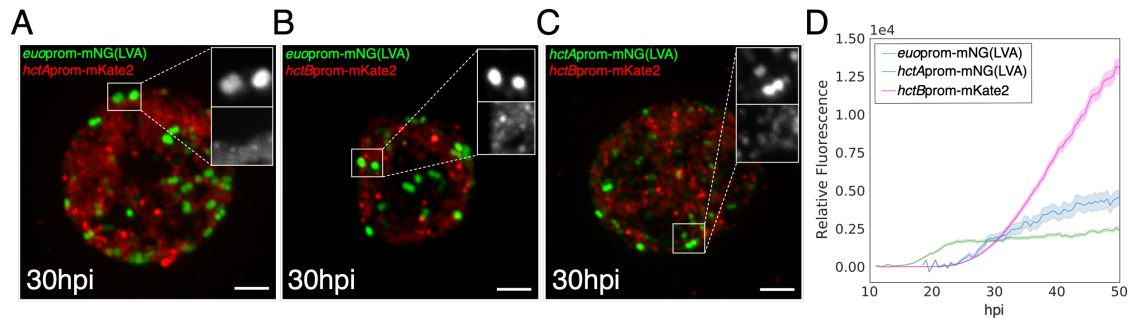


**Figure 3.1: Active cell-form specific promoter-reporter chlamydial strains.**

Representative confocal micrographs of Cos-7 cells infected with (A) *Ctr*-L2-*euo*prom-mNG(LVA)_*hctA*prom-mKate2, (B) *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 or (C) *Ctr*-L2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2 reporter strains. Magnified FOVs demonstrate cell-form specific expression in isolated cells (mNG(LVA): top, mKate2: bottom). Cells were fixed at 30 hpi. Scale bar = 10 μm. (D) Live-cell expression kinetics of *euo*prom-mNG(LVA): green, *hctA*prom-mNG(LVA): blue, and *hctB*prom-mKate2: pink, from >50 individual inclusions. Infections were monitored from 10-50 hpi via automated live-cell fluorescence microscopy. Average intensities are shown, cloud represents SEM.

**Figure 3.2.**



A  *euo*prom-mNG(LVA) positive cells

B  *hctA*prom-mNG(LVA) positive cells
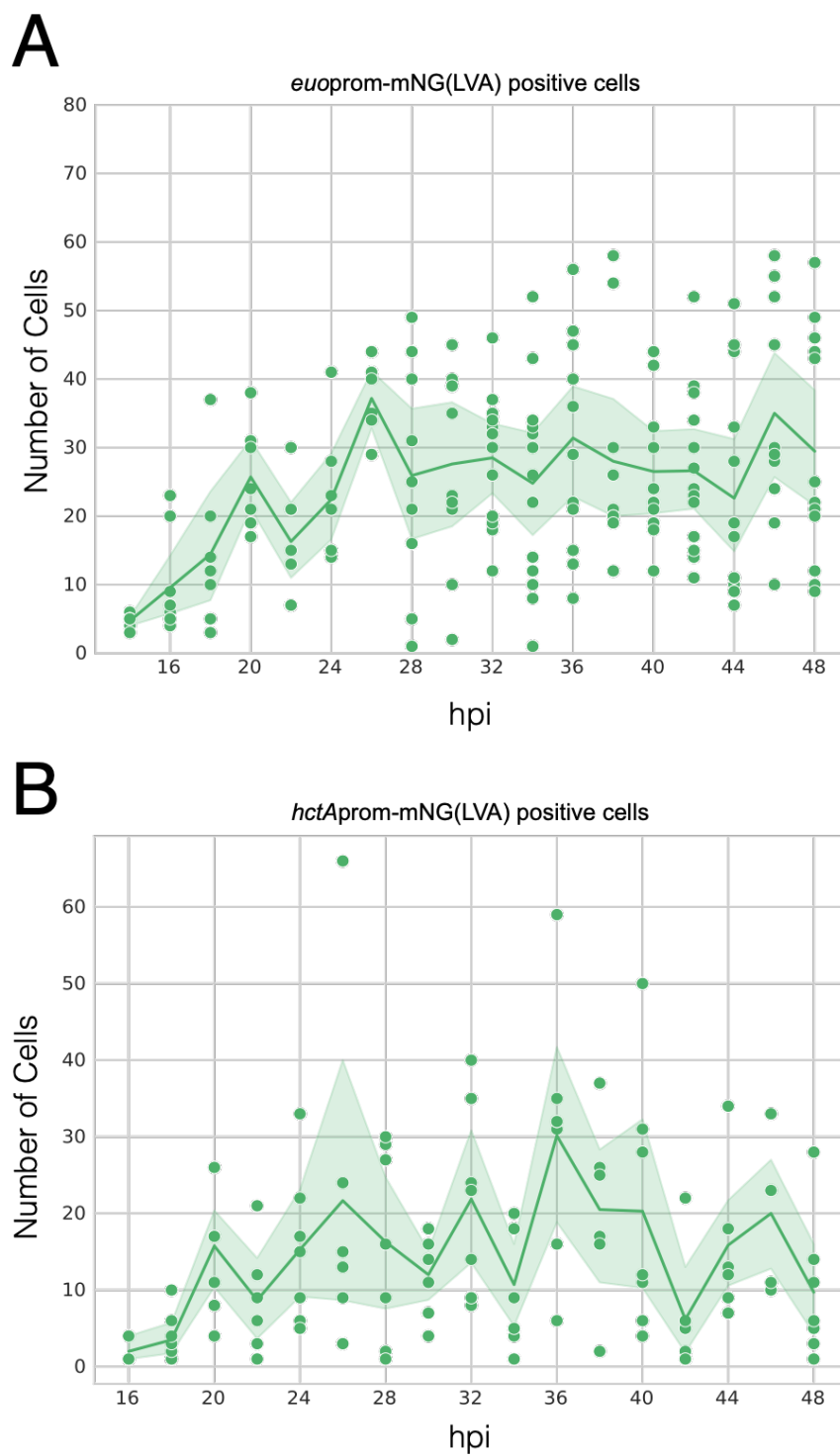
**Figure 3.2: Intra-inclusion RB and IB numbers increase and plateau.**

Cos-7 cells were infected with purified *Ctr*-L2-prom EBs. Infections were fixed every two hours from 14-48 hpi and stained with DAPI. Promoter-reporter+/DAPI+ cell were counted on a per inclusion. (A and B) Total number of *euo*prom-mNG(LVA)+ or *hctA*prom-mNG(LVA)+ cells per inclusion, respectively. Individual dots represent individual inclusions. Solid line represents the mean number of promoter-reporter+/ DAPI+ cells per time point. Sample size ranged between 3-14 inclusions. Cloud represents 95% ci.

**Figure 3.3.**
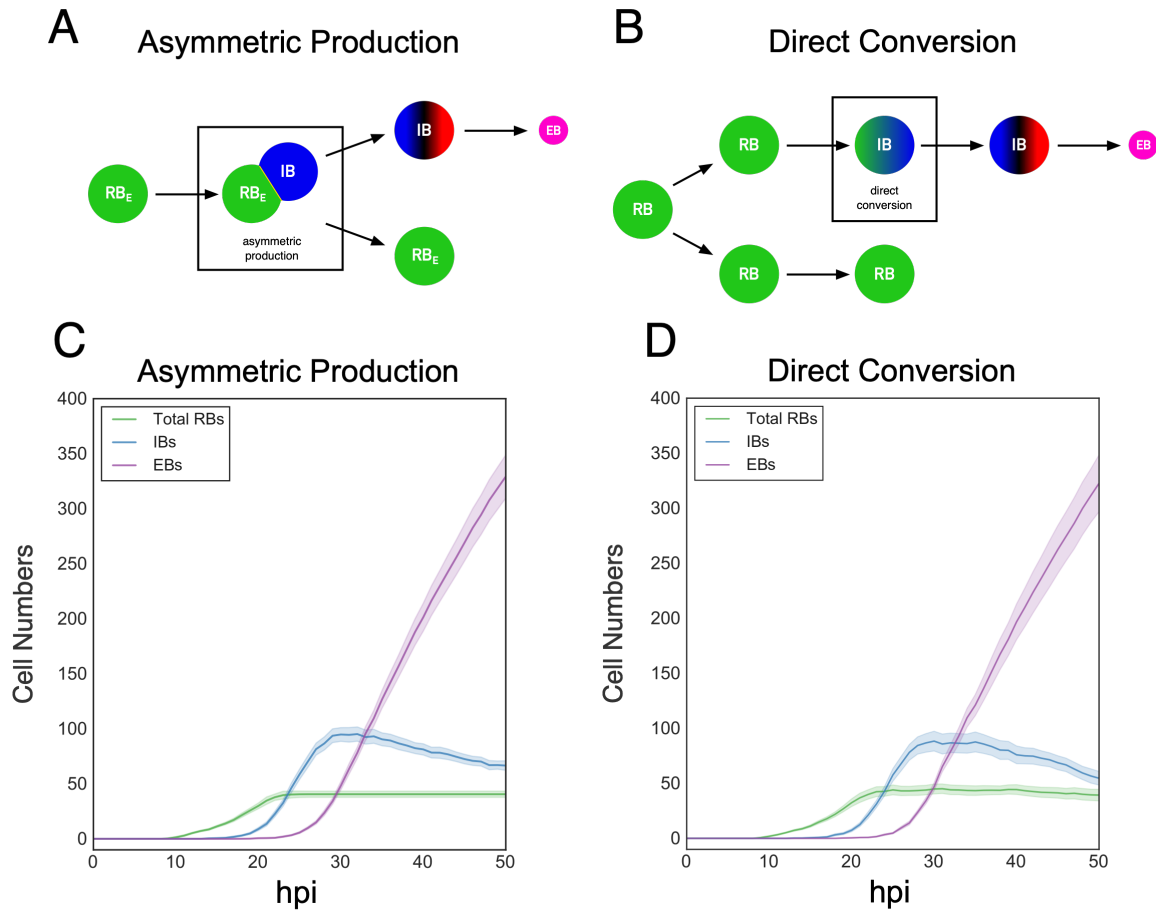


**Figure 3.3: Schematic and simulation outputs of the asymmetric production and direct conversion models.**

(A) Schematic of the asymmetric production model. Upon each $RB_E$ division event, one $RB_E$ and IB daughter cell is produced via asymmetric cell division. (B) Schematic of the direct conversion model. After each cell division event RBs 'decide' to convert directly into IBs. (C-D) Simulated cell-form subpopulation kinetic outputs of the asymmetric production and direct models, respectively. Total RBs ($RB_R$s + $RB_E$s): green, IBs: blue, and EBs: purple. Infections were simulated from 0-50 hpi. Average cell-form subpopulation numbers of 20 simulations/model are shown, cloud represents SEM. Model parameters can be found in **Supplemental Material 4.S4 and 4.S5**.

**Figure 3.4.**
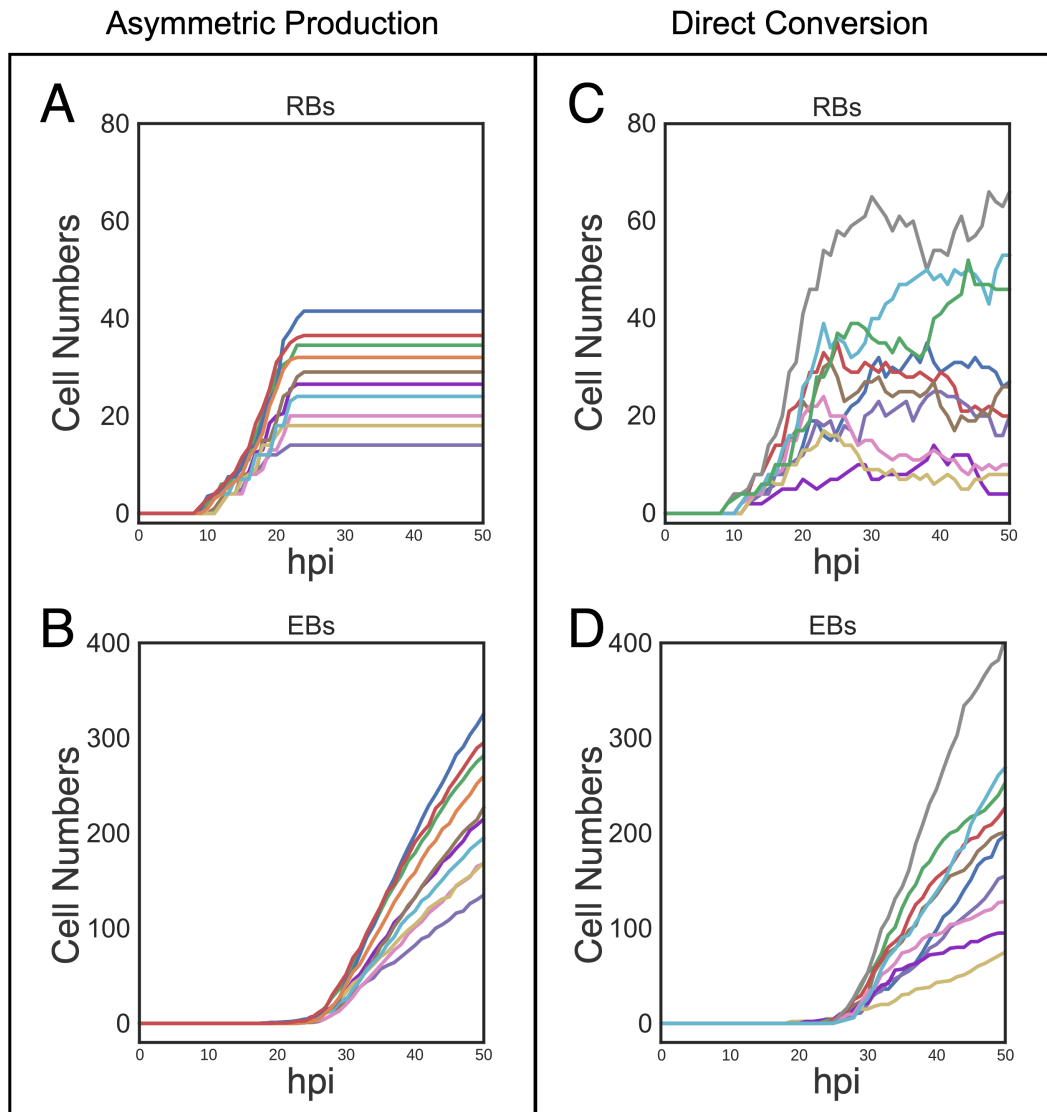


**Figure 3.4: Simulated single-inclusion kinetics of RB and EB cell numbers.**
Individual traces of simulated RB and EB kinetics on a per-inclusion level for the asymmetric production (A-B) and direct conversion (C-D) models. Inclusions were simulated from 0-50 hpi. Individual traces correspond to individual inclusion simulations. Colors of individual inclusion traces are paired between the RB and EB cell forms per model simulation.

**Figure 3.5.**
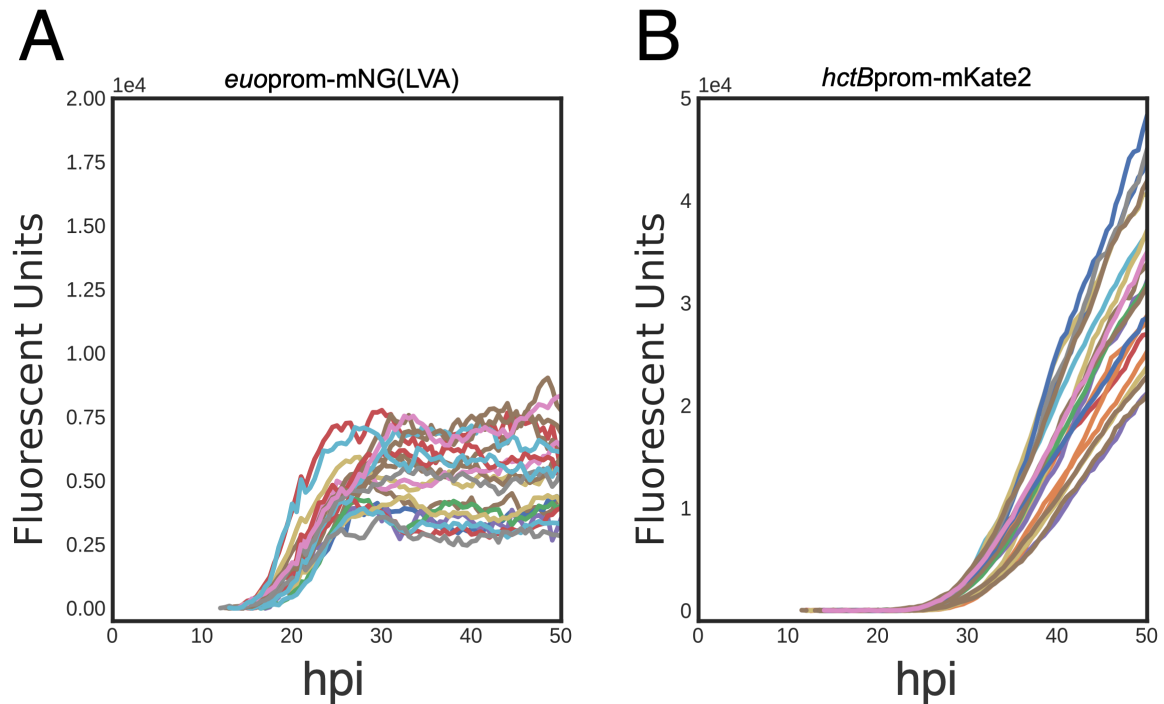


**Figure 3.5: Individual inclusion traces of RB and EB promoter-reporter kinetics.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 EBs. Individual inclusions were imaged every 30 min from 10-50 hpi**.** (A-B) Representative live-cell fluorescent kinetic traces of 20 individual inclusions expressing *euo*prom-mNG(LVA):RBs, and hctBprom-mKate2: EBs, respectively.

**Figure 3.6.**

A



B
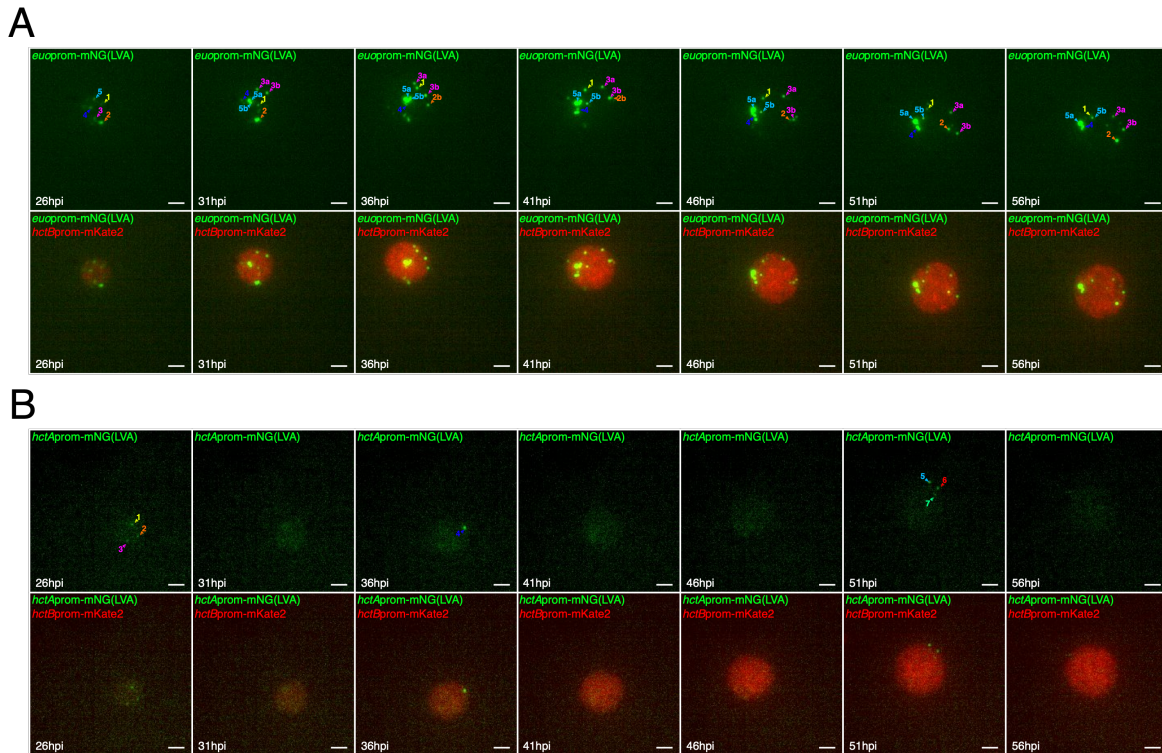


**Figure 3.6: Individual RBs are stable and IBs are transient throughout infection.**

Cos-7 cells were infected with purified *Ctr*-L2-prom EBs. Individual inclusions were imaged every 15 min from 24-60 hpi during active infections**.** (A-B) Representative 5-hour interval timelapse images of *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 (RB/EB) or *Ctr*-L2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2 (IB/EB), respectively. Numbered arrowheads indicate individual *euo*prom-mNG(LVA)+ (RB) or *Ctr*-L2-*hctA*prom-mNG(LVA)+ (IB) cells through time. Supplemental video **Mov. 4.S3** and **Mov. 4.S4** of 15 min interval timelapse for each inclusion.

**Figure 3.7.**
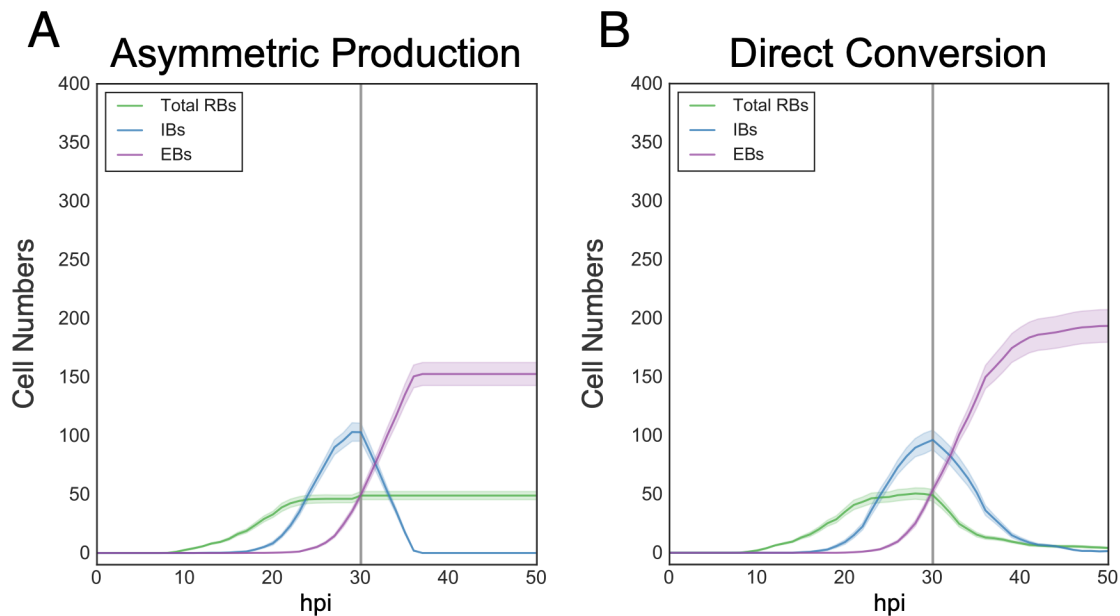


**Figure 3.7: Predicted cell-form specific outcomes in cell-division inhibition models.**

Simulated cell-form subpopulation kinetic outputs of cell-division inhibition in the asymmetric production and direct conversion model. Total RBs ($RB_{RS}$ + $RB_{ES}$): green, IBs: blue, and EBs: purple (A) Simulated outputs from the asymmetric production model. (B) Simulated outputs from the direct conversion model. Infections were simulated from 0-50 hpi. Gray vertical line indicates time of simulated cell-division inhibition. Average cell-form subpopulation numbers of 20 simulations per model are shown, cloud represents SEM.

**Figure 3.8.**
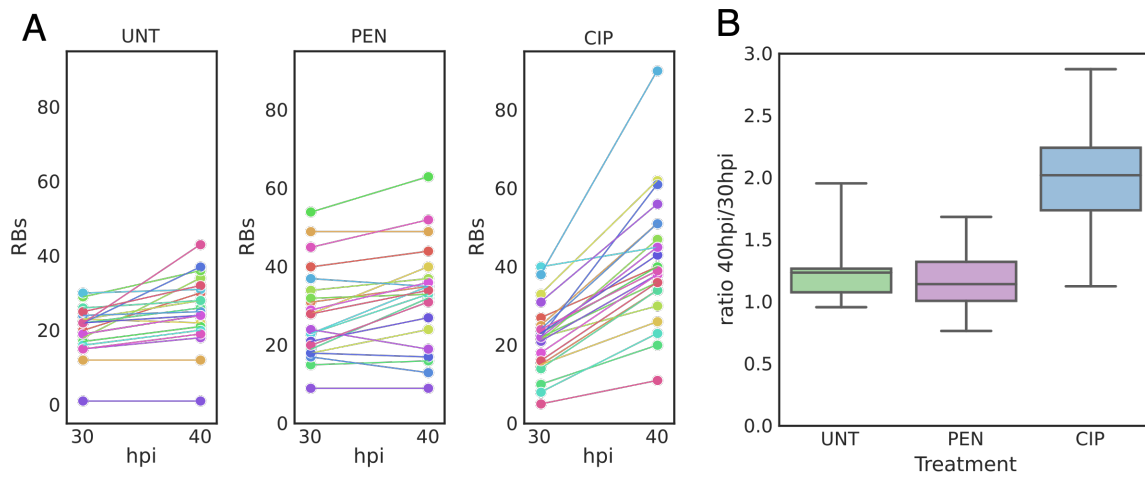


**Figure 3.8: RBs do not convert to IBs after cell division inhibition.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-mNG(LVA) EBs. Infected cells were treated at 30 hpi with either mock (UNT), penicillin-G (PEN) or ciprofloxacin (CIP). (A) The number of *euo*prom-mNG(LVA)+ cells per inclusion from live-cell experiments at 30 and 40 hpi. Individual inclusions are connected via horizontal lines. (B) The 40/30 hpi ratio of *euo*prom-mNG(LVA)+ cell numbers per treatment.

**Figure 3.9.**
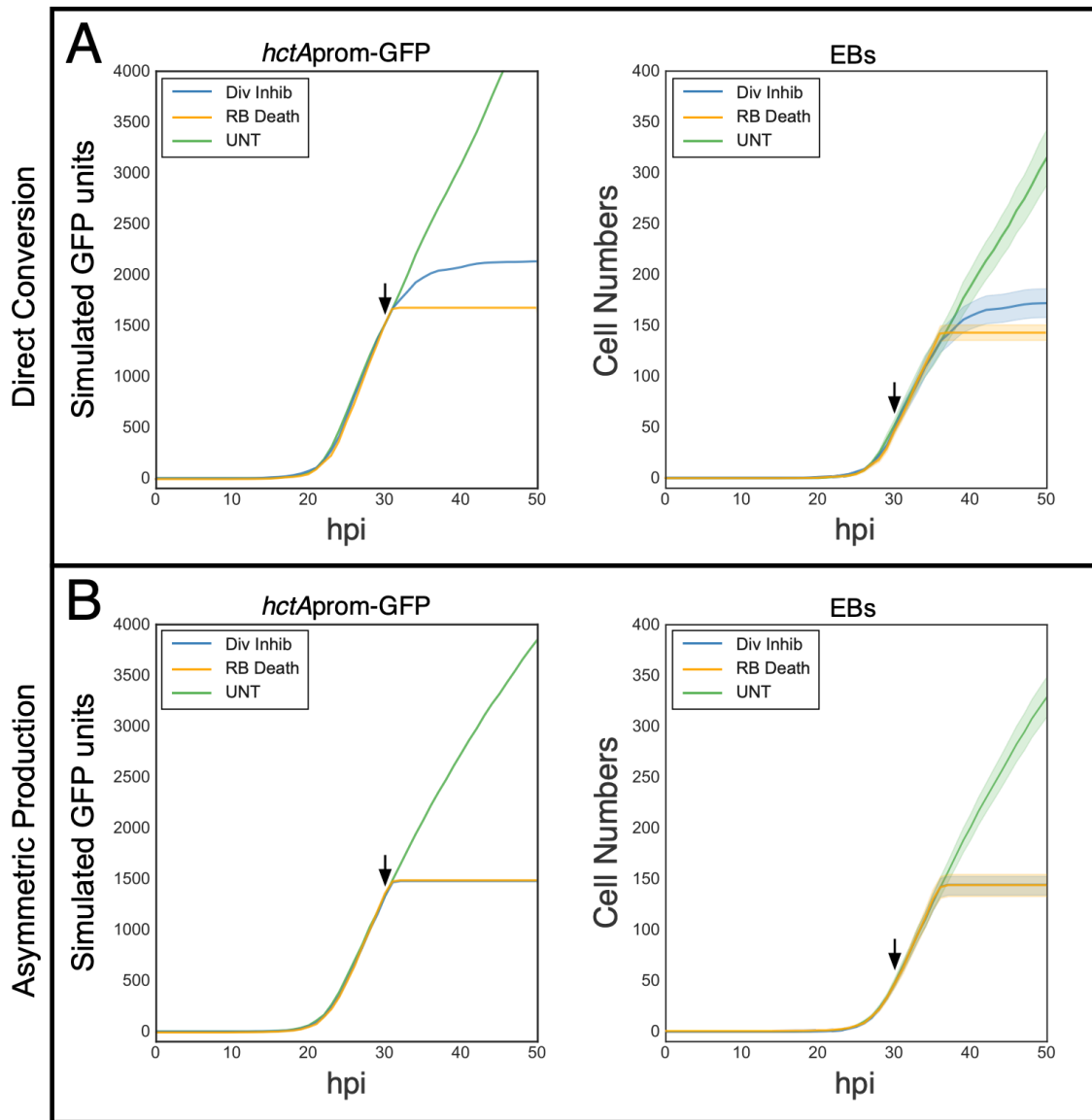
**Figure 3.9: Predicted outcomes of a photostable IB promoter-reporter and EB production post cell-division inhibition.**

Simulated kinetic outputs of total accumulation of a photostable promoter-reporter *hctA*prom-GFP and total EB cell numbers. Untreated simulation: green, cell-division inhibition: blue, RB death: orange. (A) Outputs from the asymmetric production model in untreated, cell-division inhibited and RB death simulations. (B) Outputs from the direct conversion model in untreated, cell-division inhibited and RB death simulations. Infections were simulated from 0-50 hpi. Arrow indicates time of simulated cell division inhibition. Average cell-form subpopulation numbers of 20 simulations per model are shown, normalized to the untreated production slopes. Cloud represents SEM.

**Figure 3.10.**
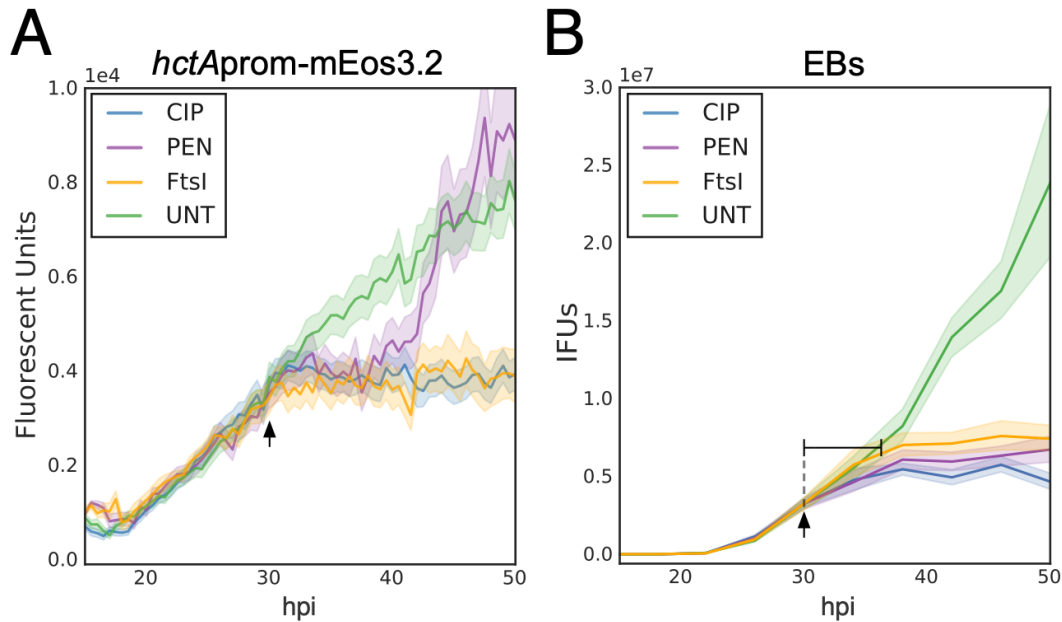


**Figure 3.10: IB production halts after cell division inhibition.**

Cos-7 cells were infected with *Ctr*-L2-*ftsI*3XFLAG_*hctA*prom-mEos3.2_*hctB*prom-mKate2 EBs. Infected cells were treated at 30 hpi with mock (UNT): green, penicillin-G (PEN): purple, ciprofloxacin (CIP): blue, or induced for FtsI: orange. (A) Live-cell expression kinetics of hctAprom-mEos3.2. (B) Quantification of EBs was performed by replating assay (IFU). Infected monolayers were harvested at 4 hour intervals from 10 to 50 hpi. Arrow indicates time of treatment. Horizontal solid line indicates the time to maximum expression from treatment. Live-cell fluorescent intensity and IFU means are shown. Averages are shown. The fluorescent unit cloud represents SEM. IFU cloud represents 95% ci.

**Figure 3.11.**



A — 20hpi

*euo*prom-mNG(LVA) DAPI | *hctB*prom-mKate2 DAPI | merge

B — 30hpi

UNT, CIP, PEN

*euo*prom-mNG(LVA) DAPI | *hctB*prom-mKate2 DAPI | merge
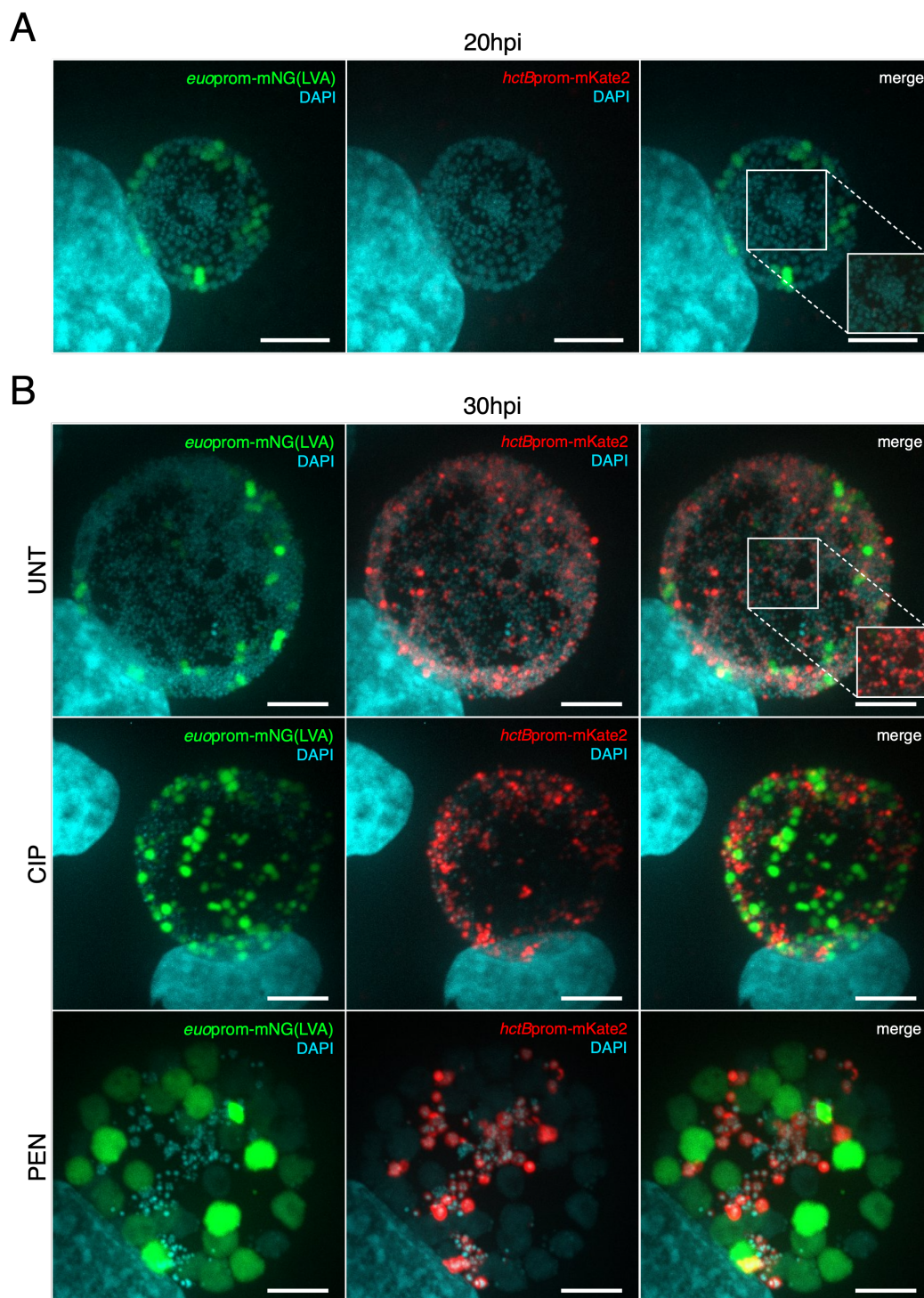
**Figure 3.11: IB-to-EB development is replication independent.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2. Infected cells were treated at 20 hpi with either mock (UNT), ciprofloxacin (CIP), or penicillin-G (PEN). Samples were fixed at 20 hpi (pre-treatment) or 30 hpi, and stained with DAPI. *euo*prom-mNG(LVA): green, *hctB*prom-mKate2: red, and DAPI: cyan. (A) Representative confocal micrograph of a 20 hpi mock-treated infection. (B) Representative confocal micrographs of 30 hpi UNT, CIP, and PEN-treated infections. Cutout FOVs are enhanced for *hctB*prom-mKate2 detection to demonstrate the absence of *hctB*prom-mKate2 expression in the 20 hpi sample and presence in the 30 hpi sample. Scale bar = 10 μm.
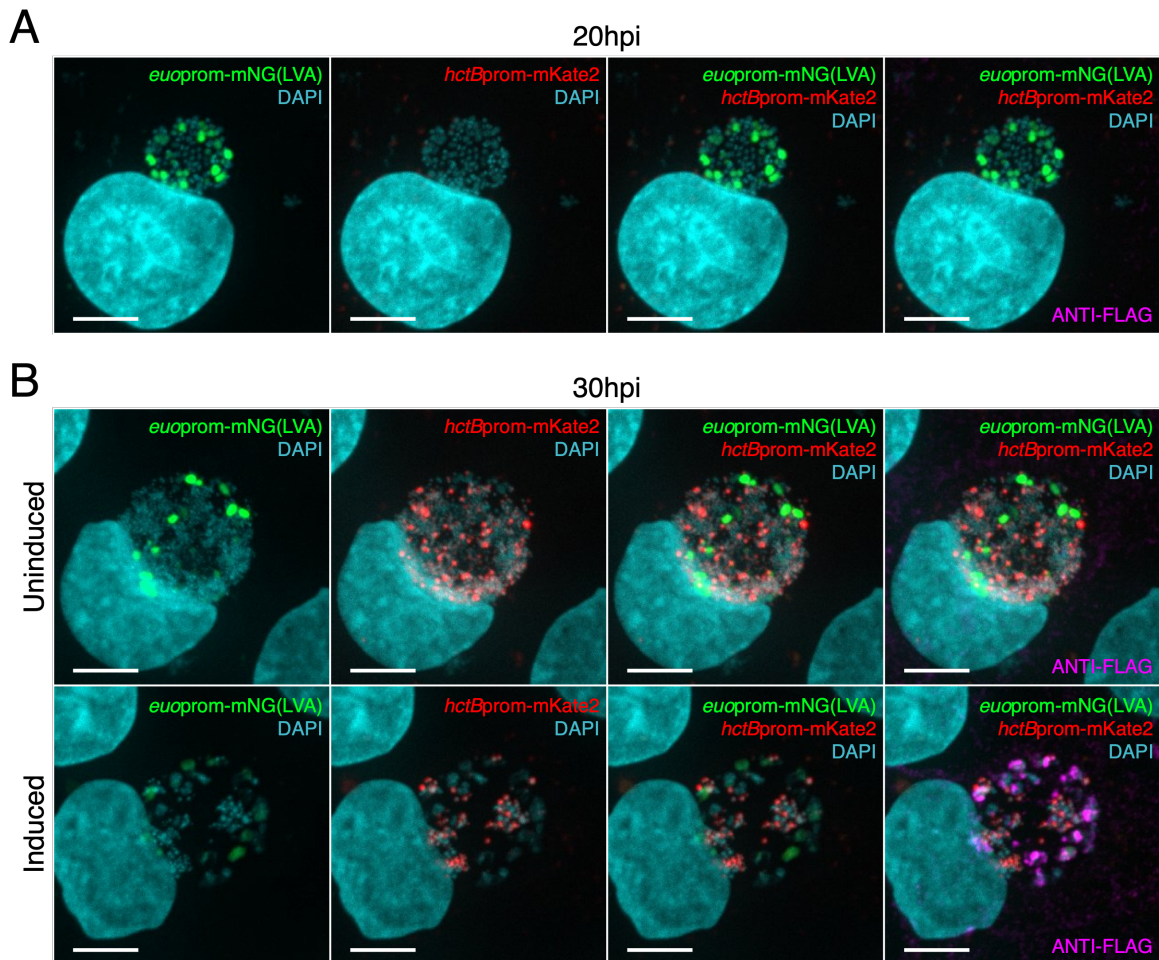
**Figure 3.12.**



**Figure 3.12: IB-to-EB development occurs post RB death.**

Cos-7 cells were infected with purified *Ctr*-L2-E-*ftsI*3XFLAG_*euo*prom-mNG(LVA)_*hctB*prom-mKate2. Infected cells were induced for FtsI3XFLAG expression at 20 hpi. Samples were fixed at 20 hpi (pre-treatment) or 30 hpi, and stained with DAPI and an anti-FLAG antibody for immunofluorescence (IF) imaging. *euo*prom-mNG(LVA): green, *hctB*prom-mKate2: red, DAPI: cyan, anti-FLAG: magenta. (A) Representative confocal micrograph of a 20 hpi mock-induced infection. (B) Representative confocal micrographs of 30 hpi uninduced and FtsI-induced infections. Scale bar = 10 µm.
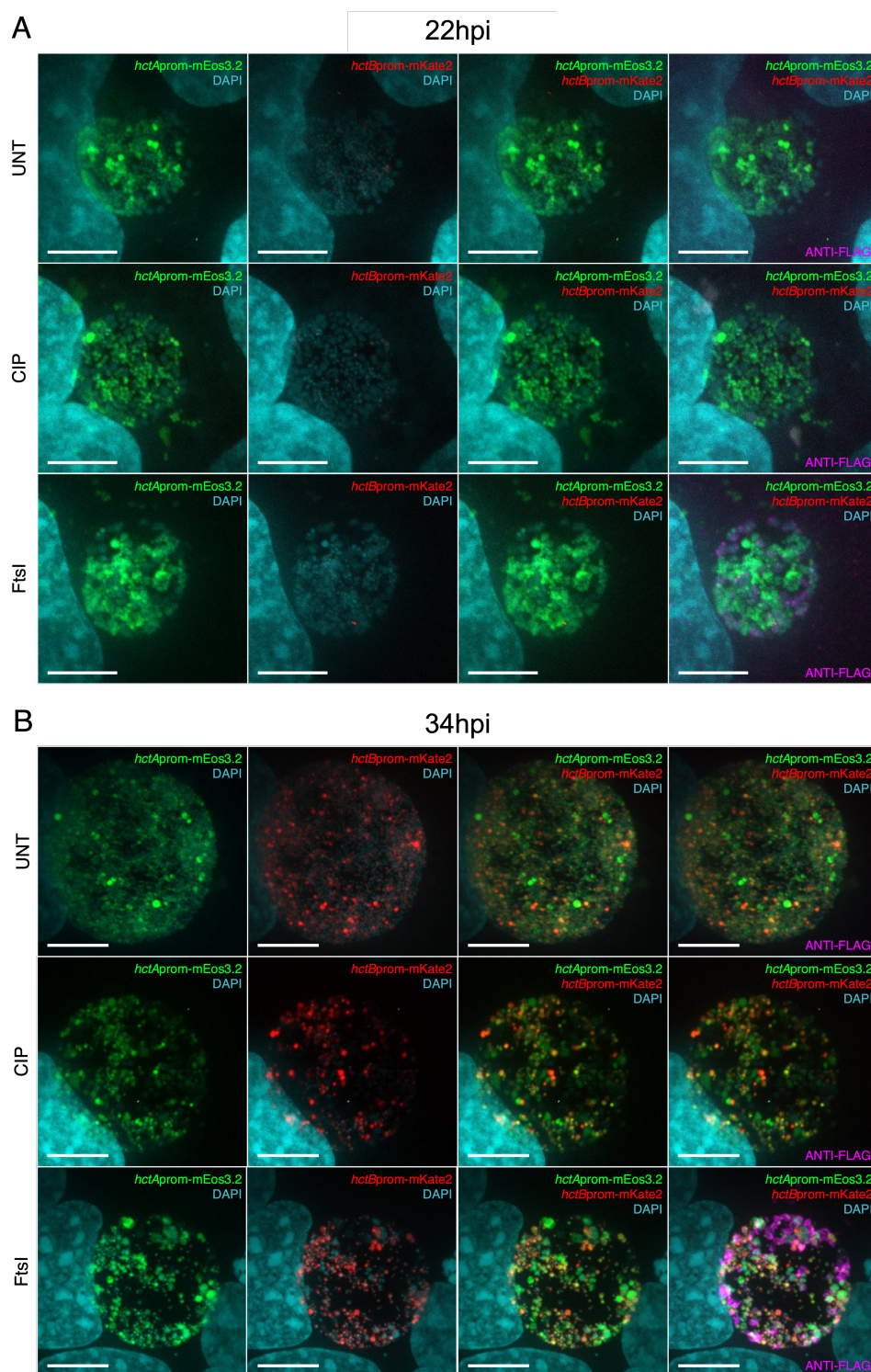
**Figure 3.13.**

**Figure 3.13: IBs mature directly into EBs post cell division.**

Cos-7 cells were infected with *Ctr*-L2-E-*ftsI*3XFLAG_*hctA*prom-mEos3.2_*hctB*prom-mKate2 EBs. Infected cells were treated at 18 hpi with either mock (UNT), ciprofloxacin (CIP), or induced for FtsI3XFLAG expression (FtsI). Samples were fixed at 22 hpi (pre-treatment) or 34 hpi, and stained with DAPI and an anti-FLAG antibody for IF imaging. *hctA*prom-mEos3.2: green, *hctB*prom-mKate2: red, DAPI: cyan, anti-FLAG: magenta. (A) Representative confocal micrographs of 22 hpi UNT, CIP, and FtsI induced samples. (B) Representative confocal micrographs of 34 hpi UNT, CIP, and FtsI induced samples. Scale bar = 10 μm.

**Figure 3.S1.**



**A** Decision < Division

**B** Decision > Division
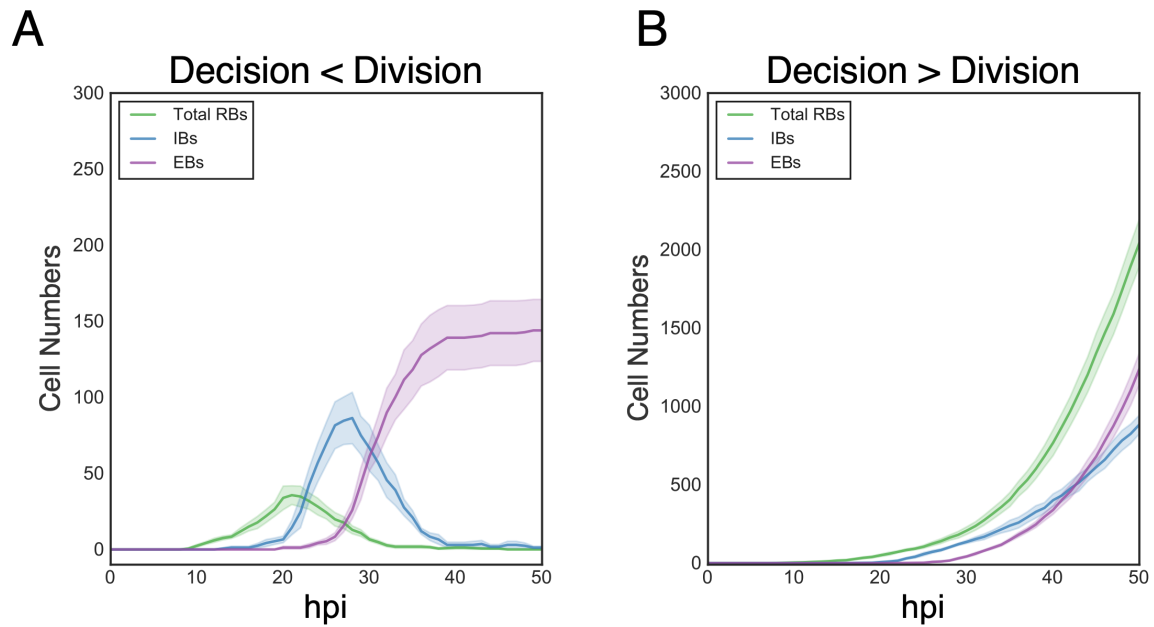
**Figure 3.S1: Simulation outputs of decreased and increased RB-to-IB decision time in direct conversion models.**

Simulated cell-form subpopulation kinetic outputs of modified direct conversion models. Total RBs (RB$_R$s + RB$_E$s): green, IBs: blue, and EBs: purple. (A) If the RB-to-IB decision time is half the time to cell division, the RB population converts to IBs at a rate faster than RB replication, leading to RB extinction. (B) If the RB-to-IB decision time is twice that of the cell division time, RB replication outcompetes IB conversion and RB population runaway occurs. Infections were simulated from 0-50 hpi. Average cell-form subpopulation numbers of 10 simulations per model are shown, cloud represents SEM.

**Figure 3.S2.**
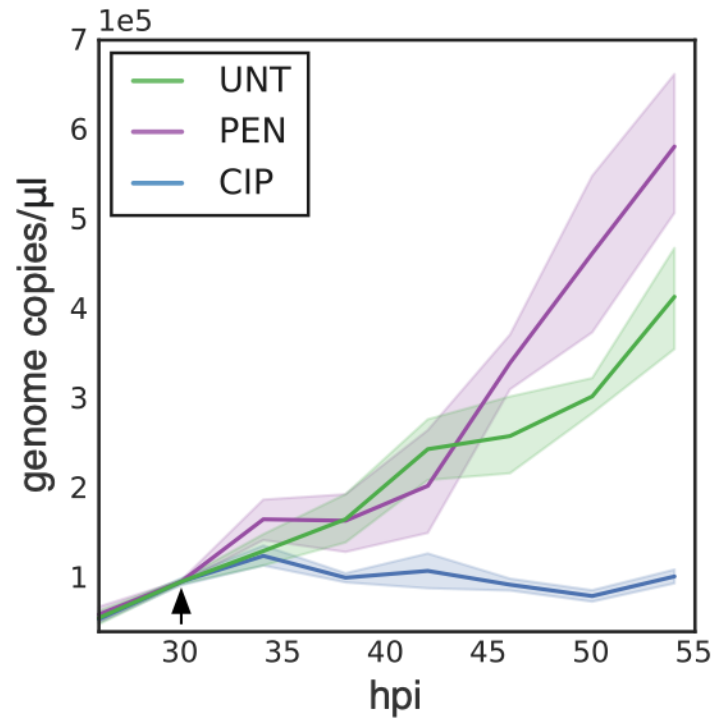


**Figure 3.S2: DNA replication is inhibited by ciprofloxacin-treatment.**

Cos-7 cells were infected with *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 EBs. Infected cells were treated at 30 hpi mock (UNT): green, penicillin-G (PEN): purple, or ciprofloxacin (CIP): blue. Genome copies were quantified using ddPCR. Samples were harvested every 4 hours from 26-54 hpi. Arrow indicates treatment time. Means are shown. Genome copy cloud represents 95% ci.

**Figure 3.S3.**
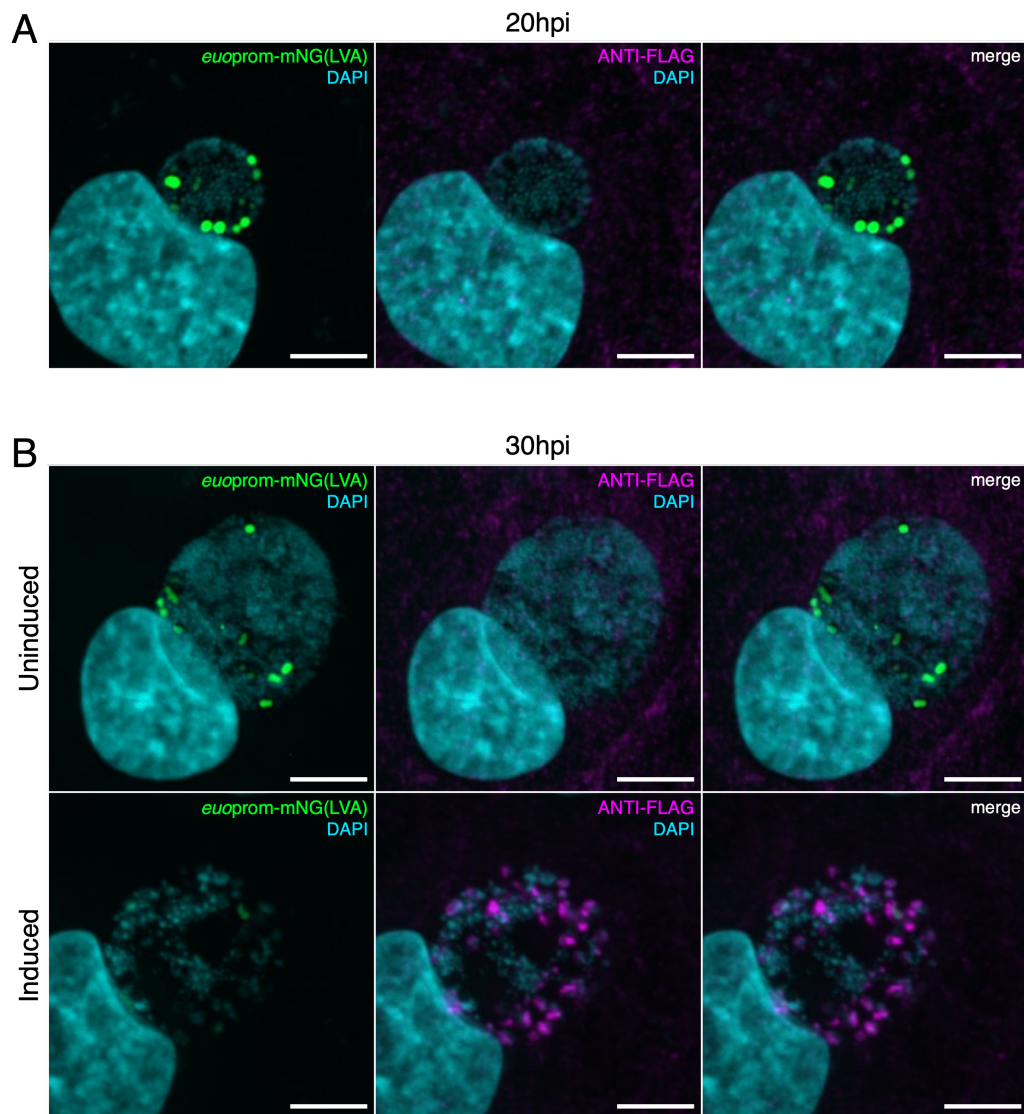


**Figure 3.S3: FtsI overexpression induces RB cell death.**

Cos-7 cells were infected with purified *Ctr*-L2-E-*ftsI*3XFLAG_*euo*prom-mNG(LVA)_*hctB*prom-mKate2. Infected cells were induced for FtsI3XFLAG expression at 20 hpi. Samples were fixed at 20 hpi (pre-treatment) or 30 hpi, and stained with DAPI and an anti-FLAG antibody for immunofluorescence (IF) imaging. *euo*prom-mNG(LVA): green, DAPI: cyan, anti-FLAG: magenta. (A) Representative confocal micrograph of a 20 hpi mock-induced infection. (B) Representative confocal micrographs of 30 hpi uninduced and induced infections. Scale bar = 10 µm.
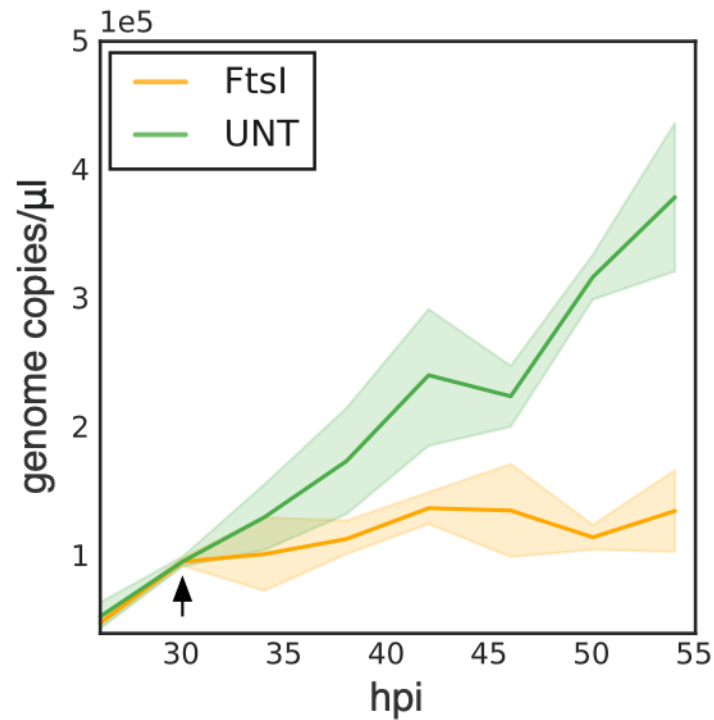
**Figure 3.S4.**



**Figure 3.S4: FtsI overexpression inhibits DNA replication.**

Cos-7 cells were infected with *Ctr*-L2-E-*ftsI*3XFLAG_*euo*prom-mNG(LVA)_*hctB*prom-mKate2 EBs. Infected cells were treated at 30 hpi with either mock (UNT): green, or induced for FtsI: orange. Genome copies were quantified using ddPCR. Samples were harvested every 4 hours from 26-54 hpi. Arrow indicates treatment time. Means are shown. Genome copy cloud represents 95% ci.

**Movie 3.S1: Simulation of the asymmetric production ABM.**

Simulated individual cell-form development of a single inclusion. Germinating EB: lavender, RBs: green, IBs: blue>black>red, and EBs: pink. The green circle follows the progression of three RBs. Individual RBs are stable and demonstrate a stem cell-like quality in IB production.

**Movie 3.S2: Simulation of the direct conversion ABM.**

Simulated individual cell-form development of a single inclusion. Germinating EB: lavender, RBs: green, IBs: blue>black>red, and EBs: pink. The green circle follows the progression of multiple individual RBs. Individual RBs are transient as they convert into IBs. However, RB division is matched with RB-to-IB conversion, leading to a fluctuating steady-state in RB numbers.

**Movie 3.S3: Live-cell time-lapse movie of individual RB stability throughout an active infection.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2 EBs. 40X automated live-cell fluorescence microscopy was used to monitor individual RBs (*euo*prom-mNG(LVA)+ cells) within single inclusions every 15 minutes from 24-60 hpi. The *hctB*prom-mKate2 promoter-reporter was used for inclusion identification and to monitor inclusion development.

**Movie 3.S4: Live-cell time-lapse movie of individual transient IBs throughout an active infection.**

Cos-7 cells were infected with purified *Ctr*-L2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2 EBs. 40X automated live-cell fluorescence microscopy was used to monitor individual IBs (*hctA*prom-mNG(LVA)+ cells) within single inclusions every 15 minutes from 24-60 hpi. The *hctB*prom-mKate2 promoter-reporter was used for inclusion identification and to monitor inclusion development.

**Movie 3.S5: Live-cell time-lapse movie of *hctA*prom-mKate2 expression in penicillin-treated aberrant RBs.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-mNG(LVA)_*hctA*prom-mKate2 EBs. Infections were treated with penicillin at 20 hpi to induce aberrancy. Automated live-cell fluorescence microscopy was used to monitor *euo*prom-mNG(LVA) and *hctA*prom-mKate2 expression levels within single inclusions every 30 minutes from 10-48 hpi.

## Supplemental Material 3.S1: Trackmate Celltype Counts

```python
1  from fiji.plugin.trackmate.detection import BlockLogDetectorFactory
2  from fiji.plugin.trackmate.detection import LogDetectorFactory
3  from fiji.plugin.trackmate.features.spot import
…  SpotIntensityAnalyzerFactory
4  from fiji.plugin.trackmate.features.spot import
…  SpotContrastAndSNRAnalyzerFactory
5  import fiji.plugin.trackmate.tracking.sparselap.SparseLAPTrackerFactory as
…  SparseLAPTrackerFactory
6  import
…  fiji.plugin.trackmate.extra.spotanalyzer.
…  SpotMultiChannelIntensityAnalyzerFactory as
…  SpotMultiChannelIntensityAnalyzerFactory
7  from fiji.plugin.trackmate.tracking.sparselap import
…  SparseLAPTrackerFactory
8  from fiji.plugin.trackmate.tracking.oldlap import SimpleLAPTrackerFactory
9  from fiji.plugin.trackmate.tracking import LAPUtils
10 import fiji.plugin.trackmate.visualization.hyperstack.HyperStackDisplayer
…  as HyperStackDisplayer
11 import fiji.plugin.trackmate.features.FeatureFilter as FeatureFilter
12 import fiji.plugin.trackmate.features.track.TrackDurationAnalyzer as
…  TrackDurationAnalyzer
13 import fiji.plugin.trackmate.features.track.TrackSpotQualityFeatureAnalyzer
…  as TrackSpotQualityFeatureAnalyzer
14 import fiji.plugin.trackmate.SelectionModel as SelectionModel
15 import fiji.plugin.trackmate.visualization.hyperstack.HyperStackDisplayer
…  as HyperStackDisplayer
16 import fiji.plugin.trackmate.Settings as Settings
17 import fiji.plugin.trackmate.Model as Model
18 import fiji.plugin.trackmate.TrackMate as TrackMate
19 import fiji.plugin.trackmate.Spot as Spot
20 import fiji.plugin.trackmate.TrackMate
21 from ij.plugin import ChannelSplitter
22 from ij.plugin import ImageCalculator
23 from net.imglib2.img.display.imagej import ImageJFunctions
24 from java.awt.event import TextListener
25 from ij import Menus
26 from ij.gui import GenericDialog
27 from ij.io import OpenDialog
28 from ij.measure import ResultsTable
29 from ij.gui import WaitForUserDialog
30 import java.util.ArrayList as ArrayList
31 import csv
32 import os
33 import sys
34 from ij import IJ
35 from ij import ImagePlus
36
37 #IJ.run("Stack to Hyperstack...", "order=xyczt(default) channels=2 slices=1
```

```python
37…  frames=45 display=Color");
38
39   imp5 = IJ.getImage()
40
41   nChannels = imp5.getNChannels()
42   # Setup settings for TrackMate
43   settings = Settings()
44   settings.setFrom(imp5)
45
46   # Spot analyzer: we want the multi-C intensity analyzer.
47   settings.addSpotAnalyzerFactory(SpotMultiChannelIntensityAnalyzerFactory())
48
49   # Spot detector.
50   settings.detectorFactory = LogDetectorFactory()
51   settings.detectorSettings = settings.detectorFactory.getDefaultSettings()
52   settings.detectorSettings['TARGET_CHANNEL'] = 1
53   settings.detectorSettings['RADIUS'] = 1.30
54   settings.detectorSettings['THRESHOLD'] = 23.0
55
56   # Spot tracker.
57   # Configure tracker – We don't want to allow merges or splits
58   settings.trackerFactory = SparseLAPTrackerFactory()
59   settings.trackerSettings = LAPUtils.getDefaultLAPSettingsMap() # almost
…    good enough
60   settings.trackerSettings['ALLOW_TRACK_SPLITTING'] = False
61   settings.trackerSettings['ALLOW_TRACK_MERGING'] = False
62   settings.trackerSettings['LINKING_MAX_DISTANCE'] = 0.5
63   settings.trackerSettings['GAP_CLOSING_MAX_DISTANCE'] = 0.5
64   settings.trackerSettings['MAX_FRAME_GAP'] = 1
65
66   # Configure track filters
67   settings.addTrackAnalyzer(TrackDurationAnalyzer())
68   settings.addTrackAnalyzer(TrackSpotQualityFeatureAnalyzer())
69
70   #filter1 = FeatureFilter('TRACK_DURATION', 20, True)
71   #settings.addTrackFilter(filter1)
72
73   # Run TrackMate and store data into Model.
74   model = Model()
75   trackmate = TrackMate(model, settings)
76
77   ok = trackmate.checkInput()
78   if not ok:
79       sys.exit(str(trackmate.getErrorMessage()))
80
81   ok = trackmate.process()
82   if not ok:
83       sys.exit(str(trackmate.getErrorMessage()))
```

**141**

**/Users/…/…/…/dissertation_scripts_and_markdown/Trackmate_Cell_type_counts.py** Page 3/4
**Saved: 5/5/22, 10:32:24 AM** **Printed for: Travis**

```python
84
85 selectionModel = SelectionModel(model)
86 displayer =  HyperStackDisplayer(model, selectionModel, imp5)
87 displayer.render()
88 displayer.refresh()
89
90 IJ.log('TrackMate completed successfully.')
91 IJ.log('Found %d spots in %d tracks.' % (model.getSpots().getNSpots(True) ,
…  model.getTrackModel().nTracks(True)))
92
93 # Print results in the console.
94 headerStr = '%10s %10s %10s %10s %10s %10s' % ('Spot_ID', 'Track_ID',
…  'Frame', 'X', 'Y', 'Z')
95 rowStr = '%10d %10d %10d %10.1f %10.1f %10.1f'
96 for i in range( nChannels ):
97     headerStr += (' %10s' % ( 'C' + str(i+1) ) )
98     rowStr += ( ' %10.1f' )
99
100 #open a file to save results
101 myfile =
…  open('/Users/scottgrieshaber/Documents/HctALVA_counts_test/test/test.csv',
…  'wb')
102 wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
103 wr.writerow(['Spot_ID', 'Track_ID', 'Frame', 'X', 'Y', 'Z', 'Channel_1',
…  'Channel_2'])
104
105 IJ.log('\n')
106 IJ.log(headerStr)
107 tm = model.getTrackModel()
108 trackIDs = tm.trackIDs(True)
109 print(tm)
110 for trackID in trackIDs:
111     spots = tm.trackSpots(trackID)
112
113     # Let's sort them by frame.
114     ls = ArrayList(spots)
115     for spot in ls:
116         values = [spot.ID(), trackID, spot.getFeature('FRAME'),
117             spot.getFeature('POSITION_X'), spot.getFeature('POSITION_Y'),
…  spot.getFeature('POSITION_Z')]
118         for i in range(nChannels):
119             values.append(spot.getFeature('MEAN_INTENSITY%02d' % (i+1)))
120
121         IJ.log(rowStr % tuple(values))
122         l1 = (values[0], values[1], values[2], values[3], values[4],
…  values[5], values[6], values[7])
123         wr.writerow(l1)
124
```

```
125  myfile.close()
```

**Supplemental Material 3.S2: Single Cell Counts by Intensity**

LVA_cell_counts_by_cell_intensity-Markdown

May 6, 2022

```python
[ ]: %matplotlib notebook
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import numpy as np; np.random.seed(22)
import seaborn as sns; sns.set(color_codes=True)
import pandas as pd
import math
```

```python
[ ]: # Imports all csv files in folder and concatonates the data sets from trackmate.

import glob

df = pd.DataFrame()
for filename in glob.glob('data/*.csv'):
    data_01 = pd.read_csv(filename, sep=',')
    filename = filename.split('/')[1]
    print(filename)
    filename = filename.split('.')[0]
    print(filename)
    data_01['Construct'] = filename.split('_')[0]
    data_01['Time'] = filename.split('_')[1]
    data_01['Inclusion_2'] = filename.split('_')[2]
    data_01['Inclusion'] = int(filename.split('_')[2])
    data = data_01[['Frame','Track_ID', 'Inclusion', 'Inclusion_2',
 'Construct', 'Time', 'X', 'Y', 'Channel_1', 'Channel_2', 'Channel_3']]
    df = df.append(data, ignore_index=True)
df['Sample_ID'] = df.Construct + '_'+ df.Inclusion_2
df['Sample_ID'] = df.Sample_ID + '-Track-' + df.Track_ID.astype(str)
```

```python
[ ]: # Create dataframe of maximum expression of individual cells

full_max_df = None
for track in df['Sample_ID'].unique():

    this_df = df[df['Sample_ID'] == track]
    max_chan2 = this_df['Channel_2'].max()
```

```
        this_df = this_df[this_df['Channel_2'] == max_chan2]

        if full_max_df is None:
            full_max_df = this_df.copy(deep=True)
        else:
            full_max_df = pd.concat([full_max_df, this_df])
```

```
[ ]: # Save maximum expression individual cell data to pickle file.

full_max_df.to_pickle("Max_intensity_cell.pkl")
```

```
[ ]: # Import baseline subtracted pickle file into Pandas dataframe.

full_max_df = pd.read_pickle("Max_intensity_cell.pkl")
```

```
[ ]: # Set threshold for identification of RBs based on DAPI and euoprom-mNGLVA␣
     ↪expression.

EA_14 = full_max_df.query('Construct == "EA" and Time == "14h" and Channel_1␣
 ↪>3000 and Channel_2 >7000')
EA_16 = full_max_df.query('Construct == "EA" and Time == "16h" and Channel_1␣
 ↪>3000 and Channel_2 >7000')
EA_18 = full_max_df.query('Construct == "EA" and Time == "18h" and Channel_1␣
 ↪>3000 and Channel_2 >7000')
EA_20 = full_max_df.query('Construct == "EA" and Time == "20h" and Channel_1␣
 ↪>3000 and Channel_2 >7000')
EA_22 = full_max_df.query('Construct == "EA" and Time == "22h" and Channel_1␣
 ↪>3000 and Channel_2 >700')
EA_24 = full_max_df.query('Construct == "EA" and Time == "24h" and Channel_1␣
 ↪>3000 and Channel_2 >700')
EA_26 = full_max_df.query('Construct == "EA" and Time == "26h" and Channel_1␣
 ↪>3000 and Channel_2 >700')
EA_28 = full_max_df.query('Construct == "EA" and Time == "28h" and Channel_1 >1␣
 ↪and Channel_2 >1')
EA_30 = full_max_df.query('Construct == "EA" and Time == "30h" and Channel_1␣
 ↪>3000 and Channel_2 >400')
EA_32 = full_max_df.query('Construct == "EA" and Time == "32h" and Channel_1␣
 ↪>3000 and Channel_2 >400')
EA_34 = full_max_df.query('Construct == "EA" and Time == "34h" and Channel_1␣
 ↪>3000 and Channel_2 >200')
EA_36 = full_max_df.query('Construct == "EA" and Time == "36h" and Channel_1␣
 ↪>3000 and Channel_2 >400')
EA_38 = full_max_df.query('Construct == "EA" and Time == "38h" and Channel_1␣
 ↪>3000 and Channel_2 >400')
EA_40 = full_max_df.query('Construct == "EA" and Time == "40h" and Channel_1␣
 ↪>3000 and Channel_2 >500')
```

```
EA_42 = full_max_df.query('Construct == "EA" and Time == "42h" and Channel_1↵
 ↪>3000 and Channel_2 >4000')
EA_44 = full_max_df.query('Construct == "EA" and Time == "44h" and Channel_1↵
 ↪>3000 and Channel_2 >4000')
EA_46 = full_max_df.query('Construct == "EA" and Time == "46h" and Channel_1↵
 ↪>3000 and Channel_2 >4000')
EA_48 = full_max_df.query('Construct == "EA" and Time == "48h" and Channel_1↵
 ↪>3000 and Channel_2 >4000')
EA_66 = full_max_df.query('Construct == "EA" and Time == "66h" and Channel_1↵
 ↪>3000 and Channel_2 >4000')
```

```
[ ]: # Set threshold for identification of IBs based on DAPI and hctAprom-mNGLVA↵
     ↪expression.

     AB_14 = full_max_df.query('Construct == "AB" and Time == "14h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_16 = full_max_df.query('Construct == "AB" and Time == "16h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_18 = full_max_df.query('Construct == "AB" and Time == "18h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_20 = full_max_df.query('Construct == "AB" and Time == "20h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_22 = full_max_df.query('Construct == "AB" and Time == "22h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_24 = full_max_df.query('Construct == "AB" and Time == "24h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_26 = full_max_df.query('Construct == "AB" and Time == "26h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_28 = full_max_df.query('Construct == "AB" and Time == "28h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_30 = full_max_df.query('Construct == "AB" and Time == "30h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_32 = full_max_df.query('Construct == "AB" and Time == "32h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <1200')
     AB_34 = full_max_df.query('Construct == "AB" and Time == "34h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <3000')
     AB_36 = full_max_df.query('Construct == "AB" and Time == "36h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <3000')
     AB_38 = full_max_df.query('Construct == "AB" and Time == "38h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <3000')
     AB_40 = full_max_df.query('Construct == "AB" and Time == "40h" and Channel_1↵
      ↪>3000 and Channel_2 >4000 and Channel_3 <3000')
     AB_42 = full_max_df.query('Construct == "AB" and Time == "42h" and Channel_1↵
      ↪>3000 and Channel_2 >3000 and Channel_3 <3000')
```

3

```
AB_44 = full_max_df.query('Construct == "AB" and Time == "44h" and Channel_1␣
 ↪>3000 and Channel_2 >1000 and Channel_3 <12000')
AB_46 = full_max_df.query('Construct == "AB" and Time == "46h" and Channel_1␣
 ↪>3000 and Channel_2 >1000 and Channel_3 <3000')
AB_48 = full_max_df.query('Construct == "AB" and Time == "48h" and Channel_1␣
 ↪>3000 and Channel_2 >2000 and Channel_3 <3000')
AB_66 = full_max_df.query('Construct == "AB" and Time == "66h" and Channel_1␣
 ↪>3000 and Channel_2 >2000 and Channel_3 <3000')
```

```
[ ]: # Create dataframe with RB counts per inclusion per timepoint

     EA_counts = pd.DataFrame()
     exp_cond = [EA_14, EA_16, EA_18, EA_20, EA_22, EA_24, EA_26, EA_28, EA_30,
                 EA_32, EA_34, EA_36, EA_38, EA_40, EA_42, EA_44, EA_46, EA_48]
     for cond in exp_cond:
         print(cond)
         for track in cond['Inclusion_2'].unique():
             this_df = cond[cond['Inclusion_2'] == track]
             max_chan2 = this_df.shape[0]
             print(max_chan2)
             print(track)
             EA_counts = EA_counts.append({'RBs':max_chan2, 'Inclusion':track,␣
     ↪"exp_cond":cond.iloc[0].Time}, ignore_index=True)
```

```
[ ]: # Drop outlier inclusion reads

     EA_counts.drop(EA_counts.query('exp_cond == "28h" and Inclusion == "006"').
      ↪index, inplace = True)
```

```
[ ]: # Assign new dataframe

     EA_counts_clean = EA_counts
```

```
[ ]: # Extract digit from exp_cond for hpi.

     EA_counts_clean['exp_cond'] = EA_counts_clean['exp_cond'].str.extract('(\d+)',␣
      ↪expand=False)
```

```
[ ]: # Set exp_cond (hpi) as an integer

     EA_counts_clean['exp_cond'] = EA_counts_clean['exp_cond'].astype(int)
```

```
[ ]: # Plot RB counts per inclusion per time point.

     with plt.style.context('seaborn-whitegrid'):
         fig, ((ax1)) = plt.subplots(nrows=1, ncols=1)
         sns.lineplot(data=EA_counts_clean, x="exp_cond", y='RBs', ax=ax1, ci=95)
```

```
        sns.scatterplot(data=EA_counts_clean, x="exp_cond", y='RBs', ax=ax1)
ax1.set_ylim([0, 80])
ax1.set_title('EUO Positive Cell Count', fontsize=10)
ax1.set_ylabel('Cells', fontsize=10)
ax1.set_xlabel('HPI', fontsize=10)
plt.xticks(fontsize=9)
plt.yticks(fontsize=9)
plt.savefig('EUO_positive_cells.pdf')
```

```
[ ]: # Create dataframe with IB counts per inclusion per timepoint

AB_counts = pd.DataFrame()
exp_cond = [AB_14, AB_16, AB_18, AB_20, AB_22, AB_24, AB_26, AB_28, AB_30,
            AB_32, AB_34, AB_36, AB_38, AB_40, AB_42, AB_44, AB_46, AB_48]
for cond in exp_cond:
    print(cond)
    for track in cond['Inclusion_2'].unique():
        this_df = cond[cond['Inclusion_2'] == track]
        max_chan2 = this_df.shape[0]
        print(max_chan2)
        print(track)
        AB_counts = AB_counts.append({'IBs':max_chan2, 'Inclusion':track,␣
 ↪"exp_cond":cond.iloc[0].Time}, ignore_index=True)
```

```
[ ]: # Drop outlier inclusion reads

AB_counts.drop(AB_counts.loc[AB_counts['IBs']>=100].index, inplace=True)
```

```
[ ]: # Assign new dataframe

AB_counts_clean = AB_counts
```

```
[ ]: # Extract digit from exp_cond for hpi.

AB_counts_clean['exp_cond'] = AB_counts_clean['exp_cond'].str.extract('(\d+)',␣
 ↪expand=False)
```

```
[ ]: # Set exp_cond (hpi) as an integer

AB_counts_clean['exp_cond'] = AB_counts_clean['exp_cond'].astype(int)
```

```
[ ]: # Plot RB counts per inclusion per time point.

with plt.style.context('seaborn-whitegrid'):
    fig, ((ax1)) = plt.subplots(nrows=1, ncols=1)
    sns.lineplot(data=AB_counts_clean, x="exp_cond", y='IBs', ax=ax1, ci=95)
    sns.scatterplot(data=AB_counts_clean, x="exp_cond", y='IBs', ax=ax1)
```

```
ax1.set_title('HctA Positive Cell Count', fontsize=10)
ax1.set_ylabel('Cells', fontsize=10)
ax1.set_xlabel('HPI', fontsize=10)
plt.xticks(fontsize=9)
plt.yticks(fontsize=9)
plt.savefig('HctA_positive_cell.pdf')
```

**Supplemental Material 3.S3: Analysis of Model Runs**

Analysis_of_model_runs_Annotated-Markdown

May 5, 2022

```python
import numpy as np
import math
import CellModeller
import pandas as pd
import os
import glob
import matplotlib.pyplot as plt
import seaborn as sns; sns.set(color_codes=True)
import pickle
import random
from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
%matplotlib notebook
```

```python
#list of cellular properties

data = pd.read_pickle('/Users/travis/Cellmodeller/data/
  ↪RbrRBe_DegSig_PollRbeIB_IBMaturetoEB063021-21-07-02-17-42/step-00010.pickle')
cs = data['cellStates']

ids = np.array([cell.id for (id,cell) in cs.items()])
lengths = np.array([cell.length for (id,cell) in cs.items()])
pos = np.array([cell.pos for (id,cell) in cs.items()])
ctype = np.array([cell.cellType for (id,cell) in cs.items()])
ccolor = np.array([cell.color for (id,cell) in cs.items()])
cellage = np.array([cell.cellAge for (id,cell) in cs.items()])
cellradius = np.array([cell.radius for (id,cell) in cs.items()])
cellvolume = np.array([cell.volume for (id,cell) in cs.items()])
celldivisions = np.array([cell.cellAdh for (id,cell) in cs.items()])
cellends = np.array([cell.ends for (id,cell) in cs.items()])
celldir = np.array([cell.dir for (id,cell) in cs.items()])
celleffGrowth = np.array([cell.effGrowth for (id,cell) in cs.items()])
cellexcludeAttr = np.array([cell.excludeAttr for (id,cell) in cs.items()])
celldivideFlag = np.array([cell.divideFlag for (id,cell) in cs.items()])
cellgrowthRate = np.array([cell.growthRate for (id,cell) in cs.items()])
cellcts = np.array([cell.cts for (id,cell) in cs.items()])
species = np.array([cell.species for (id,cell) in cs.items()])
geneamt = np.array([cell.geneamt for (id,cell) in cs.items()])
```

```
[ ]: #imports all pickle files in designated folders and concatonates the data sets␣
     ↪from simulations.

     df1 =pd.DataFrame(columns=['time', 'simulation','init','RBr','RBe','IB',␣
     ↪'pEB','mEB','AB_RB','AB_IB'
                                        ␣
     ↪,'species0_channel','species1_channel','species2_channel','species3_channel'])
     df_ctype = pd.DataFrame(columns=['time','simulation','ctype','species0_channel'
                                        ␣
     ↪'species1_channel','species2_channel','species3_channel'])
     for sim in glob.glob('/Users/travis/Cellmodeller/data/
     ↪IBpaper_TotalRBcurve_CDIStoconCoinFlipRBIB_IBMaturetoEB111121-22-01-04-15*'):
         print('--------------------------')
         df = pd.DataFrame(columns=['time', 'simulation','init','RBr','RBe','IB',␣
     ↪'pEB','mEB','AB_RB','AB_IB'])
         df_species = pd.DataFrame(columns=['time','simulation','ctype',
                                        ␣
     ↪'species0','species1','species2','species3'])
         for filename in glob.glob(sim+'/*.pickle'):
             i = 0
             name = filename.split('/')[5]
             name = name.split('_')[3]
             name = (name.split('-')[5] + '_')
             step = filename.split('/')[6]
             step = step.split('-')[1]
             step = step.split('.')[0]
             step = int(step)
             data = pd.read_pickle(filename)
             cs = data['cellStates']
             ctype = np.array([cell.cellType for (id,cell) in cs.items()])
             geneamt = np.array([cell.geneamt for (id,cell) in cs.items()])
             cellvolume = np.array([cell.volume for (id,cell) in cs.items()])
             time = step
             cell_0 = np.count_nonzero(ctype == 0)
             cell_1 = np.count_nonzero(ctype == 1)
             cell_2 = np.count_nonzero(ctype == 2)
             cell_3 = np.count_nonzero(ctype == 3)
             cell_4 = np.count_nonzero(ctype == 4)
             cell_5 = np.count_nonzero(ctype == 5)
             cell_6 = np.count_nonzero(ctype == 6)
             cell_7 = np.count_nonzero(ctype == 7)
             geneamt0 = geneamt[0,0]
             geneamt1 = geneamt[0,1]
             geneamt2 = geneamt[0,2]
             geneamt3 = geneamt[0,3]
             print('----------------------')
```

```
        df = df.append({'time':time, 'simulation':name, 'init_channel':cell_0,
    ↪'RBr':cell_1, 'RBe':cell_2,'IB':cell_3, 'pEB':cell_4,'mEB':cell_5,'AB_RB':
    ↪cell_6,'AB_IB':cell_7,'GFP_channel':geneamt0,'euo_channel':
    ↪geneamt1,'hctA_channel':geneamt2,'hctB_channel':geneamt3} ,
    ↪ignore_index=True)
        df1 = df1.append(df)
```

```
[ ]: #Sort simulations by time.

    df_ssg = df1.sort_values(by=['time'])
```

```
[ ]: #Adjust simulation time to chlamydial cycle.

    df_ssg['time']=((df_ssg['time']/10))
```

```
[ ]: #Sum RBr and RBe for total RBs.

    df_ssg['RBs'] = df_ssg['RBr'] + df_ssg['RBe']
```

```
[ ]: #Pivot on time and simulation number.

    def pivot(in_df, channel):
        in_df_p = in_df.pivot_table(index='time', columns='simulation',
    ↪values=channel)
        in_df_p['mean'], in_df_p['std'] , in_df_p['total']= in_df_p.mean(axis=1),
    ↪in_df_p.std(axis=1),in_df_p.sum(axis=1)
        return in_df_p
```

```
[ ]: #Pivot dataframe

    RBr_p = pivot(df_ssg  ,    'RBr')
    RBe_p = pivot(df_ssg  ,    'RBe')
    RBs_p = pivot(df_ssg  ,    'RBs')
    IB__p = pivot(df_ssg  ,    'IB')
    pEB_p =  pivot(df_ssg ,    'pEB')
    EB__p = pivot(df_ssg  ,    'mEB')
    ABR_p =  pivot(df_ssg ,    'AB_RB')
    ABI_p = pivot(df_ssg  ,    'AB_IB')
    euo_p = pivot(df_ssg  ,    'euo_channel')
    hcA_p = pivot(df_ssg  ,    'hctA_channel')
    hcB_p = pivot(df_ssg  ,    'hctB_channel')
    GFP_p = pivot(df_ssg  ,    'GFP_channel')
```

```
[ ]: #Graph individual simulation traces by cell type

    with plt.style.context('seaborn-white'):
        fig, (ax1,ax2,ax3,ax4,ax5) = plt.subplots(ncols=5, nrows=1)
```

```
    (RBr_p + RBe_p + ABR_p).drop(columns=['mean','std','total']).
 ↪plot(legend=False, ax=ax1)
    (IB__p + pEB_p).drop(columns=['mean','std','total']).plot(legend=False,␣
 ↪ax=ax2)
    (EB__p).drop(columns=['mean','total','std']).plot(legend=False, ax=ax3)
    (IB__p + pEB_p).drop(columns=['mean','total','std']).plot(legend=False,␣
 ↪ax=ax4)
    (EB__p).drop(columns=['mean','total','std']).plot(legend=False, ax=ax5)

    ax1.set_ylim([-2, 80])
    ax2.set_ylim([-4, 130])
    ax3.set_ylim([-20, 400])
    ax4.set_ylim([0, 400])
    ax5.set_ylim([0, 400])

    ax1.set_xlabel('HPI')
    ax2.set_xlabel('HPI')
    ax3.set_xlabel('HPI')
    ax4.set_xlabel('HPI')
    ax5.set_xlabel('HPI')

    ax1.set_xlim([0, 50])
    ax2.set_xlim([0, 50])
    ax3.set_xlim([0, 50])
    ax4.set_xlim([0, 50])
    ax5.set_xlim([0, 50])

    ax1.set_title('RBs', fontsize=11)
    ax2.set_title('IBs', fontsize=11)
    ax3.set_title('EBs', fontsize=11)
    ax4.set_title('IBs', fontsize=11)
    ax5.set_title('EBs', fontsize=11)
    fig.set_size_inches(18, 3)
```

```
[ ]: #Plot averages of individual simulations

from matplotlib.ticker import MultipleLocator
from matplotlib.ticker import AutoMinorLocator
from matplotlib.ticker import LogLocator
c = sns.color_palette('Set1',16).as_hex()
c[1]

with plt.style.context('seaborn-white'):
    fig, (ax1,ax2) = plt.subplots(ncols=2)

    def plot_sample_1(sample, color, name, style, mstyle, fcolor, i):
```

```
        ax1.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
    ↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
    ↪markeredgecolor=color, markeredgewidth=1)
        ax1.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
    ↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
    ↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_1(RBr_p +  RBe_p , c[2],  'Total RBs' , '-', '', 'None', 1)
    plot_sample_1(IB__p    + pEB_p        , c[1],  'IB' , '-', '', 'None', 1)
    plot_sample_1(EB__p            , c[3],  'EB' , '-', '', 'None', 1)

    def plot_sample_2(sample, color, name, style, mstyle, fcolor, i):
        ax2.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
    ↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
    ↪markeredgecolor=color, markeredgewidth=1)
        ax2.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
    ↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
    ↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_2(GFP_p                 , c[0],  'GFP' , '-', '', 'None', 1)
    plot_sample_2(euo_p                 , c[1],  'euo' , '-', '', 'None', 1)
    plot_sample_2(hcA_p                 , c[2],  'hcA' , '-', '', 'None', 1)
    plot_sample_2(hcB_p                 , c[3],  'hcB' , '-', '', 'None', 1)

with plt.style.context('classic'):
    ax1.legend(loc='upper left', fontsize=10)
    ax2.legend(loc='upper left', fontsize=10)

    ax1.set_xlim([0, 50])
    ax2.set_xlim([0, 40])
    ax1.set_ylim([-10, 400])

    ax1.set_xlabel('HPI')
    ax2.set_xlabel('HPI')

    ax1.arrow(30, -10, 0, 700,   color=c[8], lw=2, alpha=1, head_width = 0)
    ax2.arrow(30, -10, 0, 700,   color=c[8], lw=2, alpha=1, head_width = 0)

    fig.set_size_inches(10, 5)
```

```
[ ]: #Create dataframe with all 3 Identifiers (celltype, CELL_ID, and species␣
    ↪association (per timepoint per sim))

    df_typespec3 = pd.
    ↪DataFrame(columns=['sim','time','celltype','cell_id','species0'
                                    ,'species1','species2','species3'])
```

```python
for sim in glob.glob('/Users/travis/Cellmodeller/data/
 ↪IBpaper_TotalRBcurve_CDIPollRbeIB_IBMaturetoEB110521-22-02-21*'):
    print('-------------------------')

    df_typespec2 = pd.
 ↪DataFrame(columns=['sim','time','celltype','cell_id','species0'
                                      ,'species1','species2','species3'])
    for filename in glob.glob(sim+'/*.pickle'):

        df = pd.DataFrame()
        df_typespec0 = pd.DataFrame()
        df_typespec1 = pd.DataFrame()

        name = filename.split('/')[5]
        name = name.split('_')[3]
        name = (name.split('-')[5] + '_')
        step = filename.split('/')[6]
        step = step.split('-')[1]
        step = step.split('.')[0]
        step = int(step)

        data = pd.read_pickle(filename)
        cs = data['cellStates']
        cell_id = pd.DataFrame([cell.id for (id,cell) in cs.items()])
        cell_id = cell_id.rename(columns={0: "cell_id"})


        ctype1 = pd.DataFrame([cell.cellType for (id,cell) in cs.items()])
        ctype1 = ctype1.rename(columns={0: "celltype"})

        df_typeid = pd.concat([cell_id ,ctype1], axis=1, sort=False)

        cgeneamt = pd.DataFrame([cell.geneamt for (id,cell) in cs.items()])
        cgeneamt = cgeneamt.rename(columns={0: "GFP_channel",1: "euo_channel",2:
 ↪ "hcA_channel",3: "hcB_channel"})
        euo = cgeneamt['euo_channel'].copy()
        hcA = cgeneamt['hcA_channel'].copy()
        hcB = cgeneamt['hcB_channel'].copy()
        GFP = cgeneamt['GFP_channel'].copy()
        df_typespec0 = pd.concat([df_typeid ,cgeneamt], axis=1, sort=False)

        name2 = pd.DataFrame().reindex_like(ctype1)
        name2 = name2.rename(columns={"celltype": "sim"})
        name2['sim'] = name

        df_typespec1 = pd.concat([df_typespec0, name2], axis=1, sort=False)
```

```
        time2 = pd.DataFrame().reindex_like(ctype1)

        time2 = time2.rename(columns={"celltype": "time"})
        time2['time'] = step

        df = pd.concat([df_typespec1, time2], axis=1, sort=False)
        df_typespec2 = pd.concat([df_typespec2,df])
    df_typespec3 = pd.concat([df_typespec3,df_typespec2])
```

[ ]: df_typespec3

[ ]: #Change cell_id to str.

df_typespec3['cell_id'] = df_typespec3['cell_id'].astype(str)

[ ]: #combine simulation and cell_id for truly unique cell_id.

df_typespec3["cell_id"] = df_typespec3["sim"] + df_typespec3["cell_id"]

[ ]: #Adjust time to chlamydial cycle.

df_typespec3['time']=((df_typespec3['time']/10))

[ ]: #Make new dataframes on specific celltype

celltype0 = df_typespec3[df_typespec3.celltype == 0]
celltype1 = df_typespec3[df_typespec3.celltype == 1]
celltype2 = df_typespec3[df_typespec3.celltype == 2]
celltype3 = df_typespec3[df_typespec3.celltype == 3]
celltype4 = df_typespec3[df_typespec3.celltype == 4]
celltype5 = df_typespec3[df_typespec3.celltype == 5]
celltype6 = df_typespec3[df_typespec3.celltype == 6]
celltype7 = df_typespec3[df_typespec3.celltype == 7]

[ ]: #Pivots on cell_id and time based on celltype.

def pivot3(in_df, channel):
    in_df_p = in_df.pivot_table(index='time', columns='cell_id', values=channel)
    in_df_p['mean'], in_df_p['std'] , in_df_p['total']= in_df_p.mean(axis=1),␣
 ↪in_df_p.std(axis=1),in_df_p.sum(axis=1)
    return in_df_p

[ ]: #pivots on cell_id within celltype and time.

cell_id_celltype0_euo_p = pivot3(celltype0  ,   'euo_channel')
cell_id_celltype1_euo_p = pivot3(celltype1  ,   'euo_channel')
cell_id_celltype2_euo_p = pivot3(celltype2  ,   'euo_channel')
```

```
cell_id_celltype3_euo_p = pivot3(celltype3 ,   'euo_channel')
cell_id_celltype4_euo_p = pivot3(celltype4 ,   'euo_channel')
cell_id_celltype5_euo_p = pivot3(celltype5 ,   'euo_channel')
cell_id_celltype6_euo_p = pivot3(celltype6 ,   'euo_channel')
cell_id_celltype7_euo_p = pivot3(celltype7 ,   'euo_channel')


cell_id_celltype0_hcA_p = pivot3(celltype0 ,   'hcA_channel')
cell_id_celltype1_hcA_p = pivot3(celltype1 ,   'hcA_channel')
cell_id_celltype2_hcA_p = pivot3(celltype2 ,   'hcA_channel')
cell_id_celltype3_hcA_p = pivot3(celltype3 ,   'hcA_channel')
cell_id_celltype4_hcA_p = pivot3(celltype4 ,   'hcA_channel')
cell_id_celltype5_hcA_p = pivot3(celltype5 ,   'hcA_channel')
cell_id_celltype6_hcA_p = pivot3(celltype6 ,   'hcA_channel')
cell_id_celltype7_hcA_p = pivot3(celltype7 ,   'hcA_channel')


cell_id_celltype0_hcB_p = pivot3(celltype0 ,   'hcB_channel')
cell_id_celltype1_hcB_p = pivot3(celltype1 ,   'hcB_channel')
cell_id_celltype2_hcB_p = pivot3(celltype2 ,   'hcB_channel')
cell_id_celltype3_hcB_p = pivot3(celltype3 ,   'hcB_channel')
cell_id_celltype4_hcB_p = pivot3(celltype4 ,   'hcB_channel')
cell_id_celltype5_hcB_p = pivot3(celltype5 ,   'hcB_channel')
cell_id_celltype6_hcB_p = pivot3(celltype6 ,   'hcB_channel')
cell_id_celltype7_hcB_p = pivot3(celltype7 ,   'hcB_channel')


cell_id_celltype0_GFP_p = pivot3(celltype0 ,   'GFP_channel')
cell_id_celltype1_GFP_p = pivot3(celltype1 ,   'GFP_channel')
cell_id_celltype2_GFP_p = pivot3(celltype2 ,   'GFP_channel')
cell_id_celltype3_GFP_p = pivot3(celltype3 ,   'GFP_channel')
cell_id_celltype4_GFP_p = pivot3(celltype4 ,   'GFP_channel')
cell_id_celltype5_GFP_p = pivot3(celltype5 ,   'GFP_channel')
cell_id_celltype6_GFP_p = pivot3(celltype6 ,   'GFP_channel')
cell_id_celltype7_GFP_p = pivot3(celltype7 ,   'GFP_channel')
```

```
[ ]: #Plot protein/GFP amounts per celltype.

with plt.style.context('seaborn-white'):
    fig, (ax1,ax2,ax3,ax4,ax5) = plt.subplots(ncols=5, nrows=1)
    (cell_id_celltype1_euo_p).drop(columns=['total','std','mean']).
 ↪plot(legend=False, ax=ax1)
    (cell_id_celltype2_euo_p).drop(columns=['total','std','mean']).
 ↪plot(legend=False, ax=ax2)
    (cell_id_celltype3_euo_p).drop(columns=['total','std','mean']).
 ↪plot(legend=False, ax=ax3)
    (cell_id_celltype3_hcA_p).drop(columns=['total','std','mean']).
 ↪plot(legend=False, ax=ax4)
    (cell_id_celltype4_hcB_p).drop(columns=['total','std','mean']).
 ↪plot(legend=False, ax=ax5)
```

```
    ax1.set_xlim([0, 45])
    ax2.set_xlim([0, 45])
    ax3.set_xlim([0, 45])
    ax4.set_xlim([0, 45])
    ax5.set_xlim([0, 45])

    ax1.set_title('RBr_euo', fontsize=11)
    ax2.set_title('RBe_euo', fontsize=11)
    ax3.set_title('IB_euo', fontsize=11)
    ax4.set_title('IB_hctA', fontsize=11)
    ax5.set_title('EB_hctB', fontsize=11)

    fig.set_size_inches(40, 3)
```

```
[ ]: #Pivots on cell_id only and time (has all celltypes each cell_id ever was).

    cell_id_GFP_p = pivot3(df_typespec3 ,   'GFP_channel')
    cell_id_euo_p = pivot3(df_typespec3 ,   'euo_channel')
    cell_id_hcA_p = pivot3(df_typespec3 ,   'hcA_channel')
    cell_id_hcB_p = pivot3(df_typespec3 ,   'hcB_channel')
```

```
[ ]: #Plot protein/GFP amounts based on individual cell_id.

    with plt.style.context('seaborn-white'):
        fig, (ax1,ax2,ax3,ax4) = plt.subplots(ncols=4, nrows=1)
        (cell_id_euo_p).drop(columns=['total','std','mean']).plot(legend=False,␣
     ↪ax=ax1)
        (cell_id_hcA_p).drop(columns=['total','std','mean']).plot(legend=False,␣
     ↪ax=ax2)
        (cell_id_hcB_p).drop(columns=['total','std','mean']).plot(legend=False,␣
     ↪ax=ax3)
        (cell_id_GFP_p).drop(columns=['total','std','mean']).plot(legend=False,␣
     ↪ax=ax4)


        ax1.set_xlim([0, 40])
        ax2.set_xlim([0, 40])
        ax3.set_xlim([0, 40])
        ax4.set_xlim([0, 40])


        ax1.set_title('euo', fontsize=11)
        ax2.set_title('hctA', fontsize=11)
        ax3.set_title('hctB', fontsize=11)
        ax4.set_title('GFP', fontsize=11)
```

9

```
      fig.set_size_inches(20, 3)
```

```
[ ]: #Plot averages of individual traces.

     with plt.style.context('seaborn-white'):
         fig, (ax1) = plt.subplots(ncols=1)

         def plot_sample_1(sample, color, name, style, mstyle, fcolor, i):
             ax1.plot(sample.index, sample['total']*i, color, label=name, linestyle␣
     ↪= style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
     ↪markeredgecolor=color, markeredgewidth=1)
             ax1.fill_between(sample.index, sample['total']*i-(sample['std']*i)/math.
     ↪sqrt(len(sample.columns)), sample['total']*i+(sample['std']*i)/math.
     ↪sqrt(len(sample.columns)), color=color, alpha=0.2)

         plot_sample_1(cell_id_GFP_p  , c[0],  'GFP'   , '-', '', 'None', 1)


     with plt.style.context('classic'):
         ax1.legend(loc='upper left', fontsize=8)
         ax2.legend(loc='upper left', fontsize=8)

         ax1.set_xlim([0, 50])
         ax1.set_ylim([-100, 4000])

         ax1.set_title('hctA$prom$-GFP', fontsize=11)

         ax1.arrow(30, -100, 0, 10000,   color=c[8], lw=2, alpha=1, head_width = 0)
         fig.set_size_inches(5, 5)
```

## Supplemental Material 3.S4: Fit RB Maturation Curve

euoprom-mNGLVA_Fit_RB_Maturation_Curve-Markdown

May 5, 2022

```python
[ ]: %matplotlib notebook
     import matplotlib.pyplot as plt
     import matplotlib.ticker as ticker
     import numpy as np; np.random.seed(22)
     import seaborn as sns; sns.set(color_codes=True)
     import pandas as pd
     import math
```

```python
[ ]: #imports all csv files in folder and concatonates the data sets from trackmate

     import glob

     df = pd.DataFrame()
     for filename in glob.glob('data/*.csv'):
         data_01 = pd.read_csv(filename, sep=',')
         filename = filename.split('/')[1]
         filename = filename.split('.')[0]
         well = filename.split('_')[6]
         data_01['Well'] = well.split('-')[0]
         data_01['FOV'] = filename.split('_')[7]
         data = data_01[['Frame','Track_ID', 'Well', 'FOV', 'X', 'Y', 'Channel_1',␣
      ↪'Channel_2']]
         df = df.append(data,  ignore_index=True)
     df['Sample_ID'] = df.Well + '-' + df.FOV
     df['Sample_ID'] = df.Sample_ID + '-Track-' + df.Track_ID.astype(str)
```

```python
[ ]: #subtract baseline new using min value: Channel_1.

     def subtract_bl(in_df):
         traces = in_df
         traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
      ↪values='Channel_1')
         df_test2 = pd.DataFrame()
         for columns in traces_p:
             minvalue = traces_p[columns].min()
             base_sub = lambda x: x-minvalue
             df_test = in_df[in_df['Sample_ID']==columns]
```

1

```
        df_test['bc_channel_2'] = df_test['Channel_1']-minvalue
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df_bl)
```

```
[ ]: #subtract baseline new using min value: Channel_2.

def subtract_bl(in_df):
    traces = in_df
    traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
 ↪values='Channel_2')
    df_test2 = pd.DataFrame()
    for columns in traces_p:
        minvalue = traces_p[columns].min()
        base_sub = lambda x: x-minvalue
        df_test = in_df[in_df['Sample_ID']==columns]
        df_test['bc_channel_1'] = df_test['Channel_2']-minvalue
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df)
```

```
[ ]: # Save baseline subtracted data to pickle file.

df_bl.to_pickle("baseLine_subtract.pkl")
```

```
[ ]: # Import baseline subtracted pickle file into Pandas dataframe.

df_bl = pd.read_pickle("baseLine_subtract.pkl")
df_f=df_bl
```

```
[ ]: # Save baseline subtracted data to csv file.

df_f.to_csv("livecell_data.csv")
```

```
[ ]: #filtering out inclusion near the edges of the field of view.

df2 = df_f[~(df_f['X']<10)]
df2 = df2[~(df2['X']>670)]
df2 = df2[~(df2['Y']<10)]
df2 = df2[~(df2['Y']>670)]
df2 = df_f
```

```
[ ]: #Calibrate Frame values from the image slices to time values of experiment.

totalFrames = 76
```

2

```
startTime = 11
interval = 0.5

frame_dict = {}
for i in range(totalFrames):
    if i == 0:
        frame = i
        frame_dict[frame] = startTime+1
    else:
        frame = i
        startTime += interval
        frame_dict[frame] = startTime+1
df2['Time'] = df2['Frame'].map(frame_dict)
```

```
[ ]: #Filter out traces that do not extend over two time points.

df_f1 = df2['Sample_ID'][df2['Time']==20]
df_f2 = df2['Sample_ID'][df2['Time']==35]
df_f3 = df_f1[df_f1.isin(df_f2)]
df3 = df2[df2['Sample_ID'].isin(df_f3)]
```

```
[ ]: # Assign treatments/strains to wells, HctB = euoprom-mNGLVA_hctBprom-mKate2.

HctB = df3[df3['Sample_ID'].str.contains("C").fillna(False)|df3['Sample_ID'].
 ↪str.contains("D").fillna(False)]
```

```
[ ]: #Filter for inclusions that exhibit sufficient expression in bc_channel_1

def filterI(in_df, threshold, time):
    filter_df=in_df[in_df['Time']==time]
    traces_p = filter_df.pivot_table(index='Time', columns='Sample_ID',␣
 ↪values='bc_channel_1')
    df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
    for columns in traces_p.columns:
        max_value = traces_p[columns].max()
        if max_value > threshold:
            df_pass = df_pass.append({'Sample_ID':columns, 'pass': True},␣
 ↪ignore_index=True)

    new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
    return new_df

HctB_f      =   filterI(HctB, 1500, 25)
```

```
[ ]: #Filter for outlier inclusions in bc_channel_1

def filterII(in_df, threshold, time):
```

3

```
    filter_df=in_df[in_df['Time']==time]
    traces_p = filter_df.pivot_table(index='Time', columns='Sample_ID',␣
 ↪values='bc_channel_1')
    df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
    for columns in traces_p.columns:
        max_value = traces_p[columns].max()
        if max_value < threshold:
            df_pass = df_pass.append({'Sample_ID':columns, 'pass': True},␣
 ↪ignore_index=True)

    new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
    return new_df

HctB_f2      =   filterII(HctB_f, 6000, 36)
```

```
[ ]: #Pivot dataframe use Time as index and Sample_ID as columns.

     def pivot(in_df, channel):
         in_df_p = in_df.pivot_table(index='Time', columns='Sample_ID',␣
      ↪values=channel)
         in_df_p['mean'], in_df_p['std'], in_df_p['HPI'] = in_df_p.mean(axis=1),␣
      ↪in_df_p.std(axis=1), in_df_p.index
         return in_df_p
```

```
[ ]: #Pivot dataframe

     Euo_HB_p                = pivot(HctB_f2 ,   'bc_channel_1')
```

```
[ ]: #Plot individual inclusion traces.

     with plt.style.context('seaborn-white'):
         Euo_HB_p.drop(columns=['std', 'mean','HPI']).plot(legend=False)
```

```
[ ]: #Plot averages of individual traces

     from matplotlib.ticker import MultipleLocator
     from matplotlib.ticker import AutoMinorLocator
     from matplotlib.ticker import LogLocator
     c = sns.color_palette('Set1',16).as_hex()
     c[1]

     with plt.style.context('seaborn-white'):
         fig, (ax1) = plt.subplots(ncols=1, nrows=1)


         def plot_sample_1(sample, color, name, style, mstyle, fcolor, i):
```

4

```
        ax1.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
        #ax2.fill_between(sample.index, sample['mean']-sample['std'],␣
↪sample['mean']+sample['std'], color=color, alpha=0.15)
        ax1.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_1(Euo_HB_p      , c[1],  'Euo HB' , '-', '', 'None', 1)


with plt.style.context('classic'):

    ax1.set_title('$euo$prom-mNG(LVA)')

    ax1.set_xlim([11, 45])

    ax1.set_ylim([-500, 7000])

    fig.set_size_inches(4, 4)

    plt.savefig('euoprom-mNGLVA.pdf')
```

```
[ ]: #Create a dataframe that contains only the mean value for each time point.

     Euo_HB_p2 = Euo_HB_p[Euo_HB_p['mean'].notna()]
```

```
[ ]: #Create a dataframe that contains the sigmoidal section of the euoprom-mNGLVA␣
     ↪curve.

     Euo_HB_p2 = Euo_HB_p2[~(Euo_HB_p2['HPI'] > 32)]
```

```
[ ]: #Create a sigmoidal function.

     def sigmoid(x, L ,x0, k, b):
         y = L / (1 + np.exp(-k*(x-x0)))+b
         return (y)
```

```
[ ]: #Match a sigmoidal function to the euoprom-mNGLVA mean data.

     from scipy.optimize import curve_fit

     %matplotlib notebook
     t = Euo_HB_p2['HPI'].values
     hr = Euo_HB_p2['mean'].values
     p0 = [max(hr), np.median(t),1,min(hr)] # this is a mandatory initial guess
```

```
popt,pcov = curve_fit(sigmoid,t,hr,p0,bounds=(0, 4500), max_nfev=30000)


x = np.linspace(10, 32, 1000)
y = sigmoid(x, *popt)

plt.plot(t, hr, 'o', label='data')
plt.plot(x,y, label='fit')
plt.xlim(9, 33)
plt.legend(loc='best')

print('Parameters:', popt)
```

```
[ ]: #Graph the sigmoidal curve using th provided parameters from above.

     import matplotlib.pyplot as plt
     import numpy as np
     import math
     L = 2.06876979e+03

     c = 6.77630536e-01
     d = 1.15814642e+02



     time = np.linspace(13, 32, 1000)
     percentchance = L/(1 + np.exp((2.15841312e+01-time)*c)) + d

     plt.plot(time, percentchance)
     plt.xlabel("time")
     plt.ylabel("percentchance(time)")
     plt.xlim(12, 32)
     plt.legend(loc='best')
     plt.savefig("Fucntion_of_Time_RBechance.pdf")
```

```
[ ]: #Scale parameters to 0-100% on y-axis for RB maturation.
     #Use these new parameters to drive RB maturation in agent-based model␣
      ↪simulation.

     import matplotlib.pyplot as plt
     import numpy as np
     import math
     L = 97.81

     c = 6.77630536e-01
     d = 2.19
```

```python
time = np.linspace(10, 32, 1000)
percentchance = L/(1 + np.exp((2.15841312e+01-time)*c)) + d

with plt.style.context('seaborn-whitegrid'):
    fig, (ax) = plt.subplots(ncols=1, nrows=1)

    ax.plot(time, percentchance)
    plt.xlabel("time")
    plt.ylabel("percentchance(time)")
    plt.xlim(9, 32)
    plt.legend(loc='best')
    plt.savefig("Fucntion_of_Time_RBechance.pdf")
```

## Supplemental Material 3.S5: Asymmetric Division Model

```python
1  import random
2  from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
3  from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
4  import numpy
5  import math
6
7  from CellModeller.Integration.CLCrankNicIntegrator import
…  CLCrankNicIntegrator
8  from CellModeller.Integration.CLEulerSigIntegrator import
…  CLEulerSigIntegrator
9  from CellModeller.Signalling.GridDiffusion import GridDiffusion
10
11
12 max_cells = 2**15
13
14 #Specify parameter for solving diffusion dynamics #Add
15 grid_size = (4, 4, 4) # grid size
16 grid_dim = (64, 8, 12) # dimension of diffusion space, unit = number of
…  grid
17 grid_orig = (-128, -14, -8) # where to place the diffusion space onto
…  simulation space
18
19
20 def setup(sim):
21
22     # Set biophysics, signalling, and regulation models
23     # jitter turns on 3d
24     # gamma controls growth inhibition from neighbors
25     biophys = CLBacterium(sim, jitter_z=False, gamma = 200000)
26
27     # add the planes to set physical  boundaries of cell growth
28     #biophys.addPlane((0,-16,0), (0,1,0), 1)
29     #biophys.addPlane((0,16,0), (0,-1,0), 1)
30     sig = GridDiffusion(sim, 1, grid_dim, grid_size, grid_orig, [10.0])
31
32     # Here we set up the numerical integration:
33     # Crank-Nicholson method:
34     integ = CLCrankNicIntegrator(sim, 1, 4, max_cells, sig,
…  boundcond='reflect')
35     # Alternative is to use the simple forward Euler method:
36     #integ = CLEulerSigIntegrator(sim, 1, 2, max_cells, sig,
…  boundcond='reflect')
37
38     # use this file for reg too
39     regul = ModuleRegulator(sim, sim.moduleName)
40     # Only biophys and regulation
41     sim.init(biophys, regul, sig, integ)
42
```

```python
43
44      # Specify the initial cell and its location in the simulation
45      sim.addCell(cellType=0, pos=(0,0,0))
46      if sim.is_gui:
47
48          # Add some objects to draw the models
49          from CellModeller.GUI import Renderers
50          therenderer = Renderers.GLBacteriumRenderer(sim)
51          sim.addRenderer(therenderer)
52          #sigrend = Renderers.GLGridRenderer(sig, integ)
53          #sim.addRenderer(sigrend) #Add
54
55      sim.pickleSteps = 10
56
57
58  def init(cell):
59
60
61      # Specify mean and distribution of initial cell size
62      cell.targetVol = 2 #* numpy.random.normal(1, 0.05) #for normally
    distributed variance (middle, std)
63
64      # Specify growth rate of cells
65      cell.growthRate = 1.0 + random.uniform(-0.05,0.05)
66      cell.parentGrowth = [0] #progenitor cell logged growthRate
67
68      cell.color = [2.0, 0.5, 1.5]#specify color of cell
69      #cell.color = [1.0,0.0,0.0] #red
70      #cell.color = [0.0,1.0,0.0] #green
71      #cell.color = [0.0,0.0,1.0] #blue
72
73      #RNA and protein
74      cell.rnaamt = [0,0,0,0] # RNA levels, used, in part, to drive geneamt
    levels
75      cell.geneamt = [0.0, 0.0, 0.0, 0.0]   #hctAprom-GFP, [1]=Euo, [2]=HctA,
    [3]=HctB
76
77      #EB to RB germination time
78      cell.germTime = [(100 + random.uniform(-20,20))] #based on livecell
    data
79
80      #RBr > RBe conversion percent
81      cell.percentchance = [0,0] #curve that drives RBr > RBe conversion
82
83      #Specify initial concentration of chemical
84      cell.species[:] = [0.0] #species is concentration
85      cell.signals[:] = [0.0]
86
```

```python
87  def specRateCL(): # Signal adds at rate k1
88      return '''
89      const float k0 = 0.0f;
90      const float d0 = 0.3f;
91      float x0 = species[0];
92      rates[0] = k0 - d0*x0;
93      '''
94      # k1 = production rate of x0
95      # d1 = degradation rate of x0
96
97  def sigRateCL(): #Add
98      return '''
99      const float k1 = 1.0f;
100     float x0 = signals[0];
101     rates[0] = k1;
102     '''
103
104 time = 0
105 def update(cells):
106     global time
107     global n0
108     maturationRate = 1.0 + random.uniform(-0.05,0.05)
109     time += 1
110     time2 = (time/10)
111
112     #Iterate through each cell and flag cells that reach target size for
    …division
113
114     # Celltypes: 0=germ_EB, 1=RBr, 2=RBe, 3=IB, 4=pre_EB, 5=EB
115
116     for (id, cell) in cells.items():
117
118         if time >= cell.germTime[0]:
119
120             #based on livecell data, percent chance of RB conversion
121             cell.percentchance[0] = (97.81/(1 +
    …numpy.exp((2.15841312e+01-((time2*maturationRate)))*6.77630536e-01)) +
    …2.19)
122
123         #pr = RNA production rate
124         #nr = RNA degradation rate
125         pr0 = 0.04
126         pr1 = 0.02
127         pr2 = 0.04
128         pr3 = 0.06
129         nr0 = 0.01
130         nr1 = 0.02
131         nr2 = 0.01
```

```python
132            nr3 = 0.024
133
134        #p = protein production rate
135        #n = protein degradation rate
136        p0 = 1.0
137        p1 = 0.5
138        p2 = 1.0
139        p3 = 0.5
140        n0 = 0.05
141        n1 = 0.08
142        n2 = 0.05
143        n3 = 0.01
144
145        if cell.volume > cell.targetVol:
146            cell.divideFlag = True
147
148        if cell.cellType == 0: #germinating EB>RB
149            cell.divideFlag = False
150            cell.growthRate = 0.0
151            if time >= cell.germTime[0]: #Become RBr if time is reached
152                cell.cellType = 1 #RBr
153                cell.growthRate = 1.0 + random.uniform(-0.05,0.05)
154                cell.parentGrowth[0] = cell.growthRate
155
156
157        if cell.cellType == 1: #RBr
158            cell.rnaamt[1] = cell.rnaamt[1] + (pr1 * cell.growthRate) -
… (nr1 * cell.rnaamt[1] * cell.growthRate) #Euo RNA
159            cell.geneamt[1] = cell.geneamt[1] + (p1 * cell.growthRate *
… cell.rnaamt[1]) - (n1 * cell.growthRate * cell.geneamt[1]) #Euo
160            cell.geneamt[0] = 0 # hctAprom-GFP
161            cell.geneamt[2] = 0 # HctA
162            cell.geneamt[3] = 0 # HctB
163            cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
164
165
166            # RB maturation signal
167            if time2.is_integer() and random.uniform(0,100) <=
… cell.percentchance[0]:
168                cell.cellType = 2 #RBe conversion
169                cell.rnaamt[1] = cell.rnaamt[1] + (pr1 * cell.growthRate) -
… (nr1 * cell.rnaamt[1] * cell.growthRate) #Euo RNA
170                cell.geneamt[1] = cell.geneamt[1] + (p1 * cell.growthRate *
… cell.rnaamt[1]) - (n1 * cell.growthRate * cell.geneamt[1]) #Euo
171                cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
172
173        #need to keep for cell division inhibition
174        #if cell.cellType == 1 and time2 >= 30: #RBr
```

```
175          #     cell.cellType = 6
176          #     cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
177          #     cell.growthRate = 0
178          #     cell.geneamt[1] = cell.geneamt[1] – (n1 * cell.geneamt[1])
…  #Euo
179          #     cell.geneamt[2] = 0 # HctA
180          #     cell.geneamt[3] = 0 # HctB
181
182
183          if cell.cellType == 2: #need to keep for cell division inhibition
184              cell.rnaamt[1] = cell.rnaamt[1] + (pr1 * cell.growthRate) –
…  (nr1 * cell.rnaamt[1] * cell.growthRate) #Euo RNA
185              cell.geneamt[1] = cell.geneamt[1] + (p1 * cell.growthRate *
…  cell.rnaamt[1]) – (n1 * cell.growthRate * cell.geneamt[1]) #Euo
186
187          #if time2 >= 30: #works well if specified as different celltype
188          #     cell.cellType = 6
189          #     cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
190          #     cell.growthRate = 0
191          #     cell.geneamt[1] = cell.geneamt[1] – (n1 * cell.geneamt[1])
…  #Euo
192          #     cell.geneamt[2] = 0 # HctA
193          #     cell.geneamt[3] = 0 # HctB
194
195
196          #need to keep for cell division inhibition
197          if cell.cellType == 6: #AB
198              cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
199              cell.growthRate = 0
200              cell.geneamt[1] = cell.geneamt[1] – (n1 * cell.geneamt[1]) #Euo
201
202
203          if cell.cellType == 3: #IB
204              cell.geneamt[1] = cell.geneamt[1] – (n1 * cell.parentGrowth[0]
…  * cell.geneamt[1]) # Euo
205              cell.rnaamt[2] = cell.rnaamt[2] + (pr2 * cell.parentGrowth[0])
…  – (nr2 * cell.rnaamt[2]) #hctA RNA
206              cell.geneamt[2] = cell.geneamt[2] + (p2 * cell.parentGrowth[0]
…  * cell.rnaamt[2]) – (n2 * cell.parentGrowth[0] * cell.geneamt[2]) #hctA
207              cell.rnaamt[0] = cell.rnaamt[0] + (pr2 * cell.parentGrowth[0])
…  – (nr2 * cell.rnaamt[0]) #hctAprom–GFP RNA
208              cell.geneamt[0] = cell.geneamt[0] + (p2 * cell.parentGrowth[0]
…  * cell.rnaamt[0]) #hctAprom–GFP no degradation
209              cell.color = [[0, 0, cell.geneamt[2]*5]] #blue fast
210              if cell.geneamt[2] >= 3.5:
211                  cell.cellType = 4 #pre_EB
212
213
```

```python
214            if cell.cellType == 4: #pre_EB
215                cell.geneamt[0] = cell.geneamt[0] #hctAprom-GFP
216                cell.geneamt[2] = cell.geneamt[2] - (n2 * cell.parentGrowth[0]
     * cell.geneamt[2]) #hctA no deg for mEos
217                cell.rnaamt[3] = cell.rnaamt[3] + (pr3 * cell.parentGrowth[0])
     - (nr3 * cell.rnaamt[3]) #hctB RNA
218                cell.geneamt[3] = cell.geneamt[3] + (p3 * cell.parentGrowth[0]
     * cell.rnaamt[3])  - (n3 * cell.parentGrowth[0] * cell.geneamt[3]) #hctB
219                cell.growthRate = 0
220                cell.color = [[cell.geneamt[3]/10, 0, cell.geneamt[3]/40 +
     cell.geneamt[2]/7]] #blue to black to pink
221                if cell.geneamt[3] >= 20: #hctB
222                    cell.cellType = 5 #infectious EB
223
224
225            if cell.cellType == 5:  #infectious EB
226                cell.geneamt[0] = cell.geneamt[0] #hctAprom-GFP no degradation
227                cell.geneamt[2] = cell.geneamt[2] - (n2 * cell.parentGrowth[0]
     * cell.geneamt[2]) #hctA deg
228                cell.rnaamt[3] = cell.rnaamt[3] + (pr3 * cell.parentGrowth[0])
     - (nr3 * cell.rnaamt[3]) #hctB RNA
229                cell.geneamt[3] = cell.geneamt[3] + (p3 * cell.parentGrowth[0]
     * cell.rnaamt[3])  - (n3 * cell.parentGrowth[0] * cell.geneamt[3]) #hctB
230                cell.color = [2.0, 0.0, 0.5] #pink
231
232
233  def divide(parent, d1, d2):
234      # Specify target cell size that triggers cell division
235      # Celltype1=RBr, Celltype2=RBe, Celltype3=IB, Celltype4=immature EB,
     Celltype5=mature EB
236
237
238      if parent.cellType == 1: # If RBr: make 2 RBrs
239          d1.cellType = 1
240          d1.targetVol = 2
241          d1.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
     0.05)
242
243          d2.cellType = 1
244          d2.targetVol = 2
245          d2.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
     0.05)
246
247
248          d1.geneamt[0] = parent.geneamt[0]/2
249          d2.geneamt[0] = parent.geneamt[0]/2
250          d1.geneamt[1] = parent.geneamt[1]/2
251          d2.geneamt[1] = parent.geneamt[1]/2
```

```
252
253
254      if parent.cellType == 2: # If RBe: make 1 RBe, 1 IB
255          d1.cellType = 2
256          d1.targetVol = 2
257          d1.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
   … 0.05)
258
259          d2.cellType = 3
260          d2.growthRate = 0
261          d2.parentGrowth[0] = parent.growthRate
262          d2.targetVol = 10
263
264          d1.geneamt[0] = parent.geneamt[0]/2
265          d2.geneamt[0] = parent.geneamt[0]/2
266          d1.geneamt[1] = parent.geneamt[1]/2
267          d2.geneamt[1] = parent.geneamt[1]/2
268
269
270
271
272
273
```

## Supplemental Material 3.S6: Direct Conversion Model

```python
1  import random
2  from CellModeller.Regulation.ModuleRegulator import ModuleRegulator
3  from CellModeller.Biophysics.BacterialModels.CLBacterium import CLBacterium
4  import numpy
5  import math
6
7  from CellModeller.Integration.CLCrankNicIntegrator import
…  CLCrankNicIntegrator
8  from CellModeller.Integration.CLEulerSigIntegrator import
…  CLEulerSigIntegrator
9  from CellModeller.Signalling.GridDiffusion import GridDiffusion
10
11
12 max_cells = 2**15
13
14 #Specify parameter for solving diffusion dynamics #Add
15 grid_size = (4, 4, 4) # grid size
16 grid_dim = (64, 8, 12) # dimension of diffusion space, unit = number of
…  grid
17 grid_orig = (-128, -14, -8) # where to place the diffusion space onto
…  simulation space
18
19
20 def setup(sim):
21
22     # Set biophysics, signalling, and regulation models
23     # jitter turns on 3d
24     # gamma controls growth inhibition from neighbors
25     biophys = CLBacterium(sim, jitter_z=False, gamma = 200000)
26
27     # add the planes to set physical  boundaries of cell growth
28     #biophys.addPlane((0,-16,0), (0,1,0), 1)
29     #biophys.addPlane((0,16,0), (0,-1,0), 1)
30     sig = GridDiffusion(sim, 1, grid_dim, grid_size, grid_orig, [10.0])
31
32     # Here we set up the numerical integration:
33     # Crank-Nicholson method:
34     integ = CLCrankNicIntegrator(sim, 1, 4, max_cells, sig,
…  boundcond='reflect')
35     # Alternative is to use the simple forward Euler method:
36     #integ = CLEulerSigIntegrator(sim, 1, 2, max_cells, sig,
…  boundcond='reflect')
37
38     # use this file for reg too
39     regul = ModuleRegulator(sim, sim.moduleName)
40     # Only biophys and regulation
41     sim.init(biophys, regul, sig, integ)
42
```

```python
43
44      # Specify the initial cell and its location in the simulation
45      sim.addCell(cellType=0, pos=(0,0,0))
46      if sim.is_gui:
47
48          # Add some objects to draw the models
49          from CellModeller.GUI import Renderers
50          therenderer = Renderers.GLBacteriumRenderer(sim)
51          sim.addRenderer(therenderer)
52          #sigrend = Renderers.GLGridRenderer(sig, integ)
53          #sim.addRenderer(sigrend) #Add
54
55      sim.pickleSteps = 10
56
57
58  def init(cell):
59
60
61      # Specify mean and distribution of initial cell size
62      cell.targetVol = 2 #* numpy.random.normal(1, 0.05) #for normally
   … distributed variance (middle, std)
63
64      # Specify growth rate of cells
65      cell.growthRate = 1.0 + random.uniform(-0.05,0.05)
66      cell.parentGrowth = [0] #progenitor cell logged growthRate
67
68      cell.color = [2.0, 0.5, 1.5]#specify color of cell
69      #cell.color = [1.0,0.0,0.0] #red
70      #cell.color = [0.0,1.0,0.0] #green
71      #cell.color = [0.0,0.0,1.0] #blue
72
73      #RNA and protein
74      cell.rnaamt = [0,0,0,0] # RNA levels, used, in part, to drive geneamt
   … levels
75      cell.geneamt = [0.0, 0.0, 0.0, 0.0]   #hctAprom-GFP, [1]=Euo, [2]=HctA,
   … [3]=HctB
76
77      #EB to RB germination time
78      cell.germTime = [(100 + random.uniform(-20,20))] #based on livecell
   … data
79
80      #RBr > RBe conversion percent
81      cell.percentchance = [0,0] #curve that drives RBr > RBe conversion
82
83      #Specify initial concentration of chemical
84      cell.species[:] = [0.0] #species is concentration
85      cell.signals[:] = [0.0]
86
```

```python
87  def specRateCL(): # Signal adds at rate k1
88      return '''
89      const float k0 = 0.0f;
90      const float d0 = 0.3f;
91      float x0 = species[0];
92      rates[0] = k0 - d0*x0;
93      '''
94      # k1 = production rate of x0
95      # d1 = degradation rate of x0
96
97  def sigRateCL(): #Add
98      return '''
99      const float k1 = 1.0f;
100     float x0 = signals[0];
101     rates[0] = k1;
102     '''
103
104 time = 0
105 def update(cells):
106     global time
107     global n0
108     maturationRate = 1.0 + random.uniform(-0.05,0.05)
109     time += 1
110     time2 = (time/10)
111
112     #Iterate through each cell and flag cells that reach target size for
    division
113
114     # Celltypes: 0=germ_EB, 1=RBr, 2=RBe, 3=IB, 4=pre_EB, 5=EB
115
116     for (id, cell) in cells.items():
117
118         if time >= cell.germTime[0]:
119
120             #based on livecell data, percent chance of RB conversion
121             cell.percentchance[0] = (97.81/(1 +
    numpy.exp((2.15841312e+01-((time2*maturationRate)))*6.77630536e-01)) +
    2.19)
122
123         #pr = RNA production rate
124         #nr = RNA degradation rate
125         pr0 = 0.04
126         pr1 = 0.02
127         pr2 = 0.04
128         pr3 = 0.06
129         nr0 = 0.01
130         nr1 = 0.02
131         nr2 = 0.01
```

176

/Users/travis/Desktop/Diss…/…/…/…/…/py_files/Direct_conversion_model.py   Page 4/7
Saved: 5/16/22, 2:23:30 PM                                     Printed for: Travis

```python
132            nr3 = 0.024
133
134            #p = protein production rate
135            #n = protein degradation rate
136            p0 = 1.0
137            p1 = 0.5
138            p2 = 1.0
139            p3 = 0.5
140            n0 = 0.05
141            n1 = 0.08
142            n2 = 0.05
143            n3 = 0.01
144
145            if cell.volume > cell.targetVol:
146                cell.divideFlag = True
147                cell.parentAge[0] = cell.cellAge/10
148
149            if cell.cellType == 0: #germinating EB>RB
150                cell.divideFlag = False
151                cell.growthRate = 0.0
152                cell.coinflip = 1
153                if time >= cell.germTime[0]: #Become RBr if time is reached
154                    cell.cellType = 1 #RBr
155                    cell.growthRate = 1.0 + random.uniform(-0.05,0.05)
156                    cell.parentGrowth[0] = cell.growthRate
157
158
159            if cell.cellType == 1: #RBr
160                cell.rnaamt[1] = cell.rnaamt[1] + (pr1 * cell.growthRate) -
   (nr1 * cell.rnaamt[1] * cell.growthRate) #Euo RNA
161                cell.geneamt[1] = cell.geneamt[1] + (p1 * cell.growthRate *
   cell.rnaamt[1]) - (n1 * cell.growthRate * cell.geneamt[1]) #Euo
162                cell.geneamt[0] = 0 # hctAprom-GFP
163                cell.geneamt[2] = 0 # HctA
164                cell.geneamt[3] = 0 # HctB
165                cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
166
167
168                # RB maturation signal
169                if time2.is_integer() and random.uniform(0,100) <=
   cell.percentchance[0]:
170                    cell.cellType = 2 #RBe conversion
171                    cell.rnaamt[1] = cell.rnaamt[1] + (pr1 * cell.growthRate) -
   (nr1 * cell.rnaamt[1] * cell.growthRate) #Euo RNA
172                    cell.geneamt[1] = cell.geneamt[1] + (p1 * cell.growthRate *
   cell.rnaamt[1]) - (n1 * cell.growthRate * cell.geneamt[1]) #Euo
173                    cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
174
```

```
175          #need to keep for cell division inhibition
176          #if cell.cellType == 1 and time2 >= 30: #RBr
177          #    cell.cellType = 6
178          #    cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
179          #    cell.growthRate = 0
180          #    cell.geneamt[1] = cell.geneamt[1] - (n1 * cell.geneamt[1])
…    #Euo
181          #    cell.geneamt[2] = 0 # HctA
182          #    cell.geneamt[3] = 0 # HctB
183
184
185          if cell.cellType == 2: #need to keep for cell division inhibition
186              cell.rnaamt[1] = cell.rnaamt[1] + (pr1 * cell.growthRate) -
…    (nr1 * cell.rnaamt[1] * cell.growthRate) #Euo RNA
187              cell.geneamt[1] = cell.geneamt[1] + (p1 * cell.growthRate *
…    cell.rnaamt[1]) - (n1 * cell.growthRate * cell.geneamt[1]) #Euo
188              if cell.coinflip == 0:
189                  cell.cellType = 3
190
191              #if time2 >= 30: #works well if specified as different celltype
192              #    cell.cellType = 6
193              #    cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
194              #    cell.growthRate = 0
195              #    cell.geneamt[1] = cell.geneamt[1] - (n1 * cell.geneamt[1])
…    #Euo
196              #    cell.geneamt[2] = 0 # HctA
197              #    cell.geneamt[3] = 0 # HctB
198
199
200          #need to keep for cell division inhibition
201          if cell.cellType == 6: #AB
202              cell.color = [[1/cell.geneamt[1], 1, 1/cell.geneamt[1]]]
203              cell.growthRate = 0
204              cell.geneamt[1] = cell.geneamt[1] - (n1 * cell.geneamt[1]) #Euo
205              if ((cell.cellAge)/10) % (cell.parentAge[0]) == 0 and
…    random.randint(0, 1) == 0:
206                  cell.cellType = 3
207
208
209          if cell.cellType == 3: #IB
210              cell.growthRate = 0
211              cell.geneamt[1] = cell.geneamt[1] - (n1 * cell.parentGrowth[0]
…    * cell.geneamt[1]) # Euo
212              cell.rnaamt[2] = cell.rnaamt[2] + (pr2 * cell.parentGrowth[0])
…    - (nr2 * cell.rnaamt[2]) #hctA RNA
213              cell.geneamt[2] = cell.geneamt[2] + (p2 * cell.parentGrowth[0]
…    * cell.rnaamt[2]) - (n2 * cell.parentGrowth[0] * cell.geneamt[2]) #hctA
214              cell.rnaamt[0] = cell.rnaamt[0] + (pr2 * cell.parentGrowth[0])
```

```
214… – (nr2 * cell.rnaamt[0]) #hctAprom–GFP RNA
215          cell.geneamt[0] = cell.geneamt[0] + (p2 * cell.parentGrowth[0]
 … * cell.rnaamt[0]) #hctAprom–GFP no degradation
216          cell.color = [[0, 0, cell.geneamt[2]*5]] #blue fast
217          if cell.geneamt[2] >= 3.5:
218              cell.cellType = 4 #pre_EB
219
220
221      if cell.cellType == 4: #pre_EB
222          cell.geneamt[0] = cell.geneamt[0] #hctAprom–GFP
223          cell.geneamt[2] = cell.geneamt[2] – (n2 * cell.parentGrowth[0]
 … * cell.geneamt[2]) #hctA no deg for mEos
224          cell.rnaamt[3] = cell.rnaamt[3] + (pr3 * cell.parentGrowth[0])
 … – (nr3 * cell.rnaamt[3]) #hctB RNA
225          cell.geneamt[3] = cell.geneamt[3] + (p3 * cell.parentGrowth[0]
 … * cell.rnaamt[3])  – (n3 * cell.parentGrowth[0] * cell.geneamt[3]) #hctB
226          cell.growthRate = 0
227          cell.color = [[cell.geneamt[3]/10, 0, cell.geneamt[3]/40 +
 … cell.geneamt[2]/7]] #blue to black to pink
228          if cell.geneamt[3] >= 20: #hctB
229              cell.cellType = 5 #infectious EB
230
231
232      if cell.cellType == 5:  #infectious EB
233          cell.geneamt[0] = cell.geneamt[0] #hctAprom–GFP no degradation
234          cell.geneamt[2] = cell.geneamt[2] – (n2 * cell.parentGrowth[0]
 … * cell.geneamt[2]) #hctA deg
235          cell.rnaamt[3] = cell.rnaamt[3] + (pr3 * cell.parentGrowth[0])
 … – (nr3 * cell.rnaamt[3]) #hctB RNA
236          cell.geneamt[3] = cell.geneamt[3] + (p3 * cell.parentGrowth[0]
 … * cell.rnaamt[3])  – (n3 * cell.parentGrowth[0] * cell.geneamt[3]) #hctB
237          cell.color = [2.0, 0.0, 0.5] #pink
238
239
240 def divide(parent, d1, d2):
241     # Specify target cell size that triggers cell division
242     # Celltype1=RBr, Celltype2=RBe, Celltype3=IB, Celltype4=immature EB,
 … Celltype5=mature EB
243
244
245     if parent.cellType == 1: # If RBr: make 2 RBrs
246         d1.cellType = 1
247         d1.targetVol = 2
248         d1.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
 … 0.05)
249
250         d2.cellType = 1
251         d2.targetVol = 2
```

```
252        d2.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
   … 0.05)
253
254        d1.geneamt[0] = parent.geneamt[0]/2
255        d2.geneamt[0] = parent.geneamt[0]/2
256        d1.geneamt[1] = parent.geneamt[1]/2
257        d2.geneamt[1] = parent.geneamt[1]/2
258
259
260     if parent.cellType == 2: # If RBe: make 2 RBes
261        d1.cellType = 2
262        d1.targetVol = 2
263        d1.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
   … 0.05)
264        d1.coinflip = random.randint(0, 1)
265
266        d2.cellType = 2
267        d2.targetVol = 2
268        d2.growthRate = parent.parentGrowth[0] * numpy.random.normal(1,
   … 0.05)
269        d2.coinflip = random.randint(0, 1)
270
271        d1.geneamt[0] = parent.geneamt[0]/2
272        d2.geneamt[0] = parent.geneamt[0]/2
273        d1.geneamt[1] = parent.geneamt[1]/2
274        d2.geneamt[1] = parent.geneamt[1]/2
275
276
277
278
279
```

**Table 3.1**

| Construct/Use | Primer Name | Template |
|---|---|---|
| p2TK2-*ihtA*prom-mNG(LVA)_*hctA*prom-mKate2 | | |

| | mNeonGreen(LVA) gBlock |
|---|---|
| GAAATTAAGCatggtgagcaaaggcgaagaagataacatggcgagcctgccggcgacccatgaactgcatattttttggcagcattaacggcgtggattttgatatggtgggccagggcaccggcaacccgaacgatggctatgaagaactgaacctgaaaagcaccaaaggcgatctgcagtttagcccgtggattctggtgccgcatattggctatggctttcatcagtatctgccgtatccggatggcatgagcccgtttcaggcggcgatggtggatggcagcggctatcaggtgcatcgcaccatgcagtttgaagatggcgcgagcctgaccgtgaactatcgctatacctatgaaggcagccatattaaaggcgaagcgcaggtgaaaggcaccggctttccggcggatggcccggtgatgaccaacagcctgaccgcggcggattggtgccgcagcaaaaaaacctatcgaacgataaaaccattattagcacctttaaatggagctataccaccggcaacggcaaacgctatcgcagcaccgcgcgcaccacctatacctttgcgaaaccgatggcggcgaactatctgaaaaaccagccgatgtatgtgtttcgcaaaaccgaactgaaacatagcaaaaccgaactgaactttaaagaatggcagaaagcgtttaccgatgtgatgggcatggatgaactgtataaaAGGCCTGCAGCAAACGACGAAAACT | |

| | | |
|---|---|---|
| actgtataaaAGGCCTGCAG CAAACGACGAAAACTACG | 5' LVA | **mNeonGreen(LVA) gBlock** |
| CGATTTCTAAGCAGGAAT GGACAGTTTTTTTTGAAG CGCTCCGGATAG | 3' mNG(LVA) | **mNeonGreen(LVA) gBlock** |
| CCATTCCTGCTTAGAAAT CGATTCTGTTTTGATTTTG TCTCGGATTTTAAAAAATG TAGTG | 5' mNG(LVA) vector | **p2TK2-*ihtA*prom-mNG_*hctA*prom-mKate2** |
| CTGCAGGCCTtttatacagttca tccatgcccatcacatcggtaaacgc | 3' mNG vector | **p2TK2-*ihtA*prom-mNG_*hctA*prom-mKate2** |
| **p2TK2-*euo*prom-mNG(LVA)_*hctB*prom-mKate2** | | |
| cgaattcggcatggtgagcaaaggc gaagaagataacatgg | 5' mNG(LVA) | **p2TK2-*ihtA*prom-mNG(LVA)_*hctA*prom-mKate2** |
| CGATTTCTAAGCAGGAAT GGACAGTTTTTTTTGAAG CGCTCCGGATAG | 3' mNG(LVA) | **p2TK2-*ihtA*prom-mNG(LVA)_*hctA*prom-mKate2** |
| CCATTCCTGCTTAGAAAT CGATTCTGTTTTGATTTTG TCTCGGATTTTAAAAAATG TAGTG | 5' mNG(LVA) vector | **p2TK2-*euo*prom-Clover_*hctB*prom-mKate2** |
| tgctcaccatgccgaattcgacccct gtatcttgttgtaagc | 3' *euo*prom vector | **p2TK2-*euo*prom-Clover_*hctB*prom-mKate2** |
| **p2TK2-*euo*prom-mNG(LVA)_*hctA*prom-mKate2** | | |

| | | |
|---|---|---|
| CGGCCGCGTCtatttttaacaa accacttgattaataagtttttttgttggg aaaatattacc | 5' *euo*prom(LVA) | i h t A p r o m - m N G ( L V A ) _ h c t A p r o m - mKate2 |
| CGATTTCTAAGCAGGAAT GGACAGTTTTTTTTGAAG CGCTCCGGATAG | 3' mNG(LVA) | i h t A p r o m - m N G ( L V A ) _ h c t A p r o m - mKate2 |
| CCATTCCTGCTTAGAAAT CGATTCTGTTTTGATTTTG TCTCGGATTTTAAAAAATG TAGTG | 5' m N G ( L V A ) vector | p 2 T K 2 - e u o p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2 |
| gttaaaaataGACGCGGCCG CGTCTTAGGAGCTTTTTG CAATGC | 3' *euo*prom(LVA) vector | p 2 T K 2 - e u o p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2 |
| **p 2 T K 2 - *h c t A* p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2** | | |
| CGGCCGCGTCttagattctaga aaatggttgc | 5' *hctA*prom(LVA) | p 2 T K 2 - *i h t A* p r o m - m N G ( L V A ) _ *h c t A* p r o m - mKate2 |
| tgctcaccatCGgatccgcCATT TTTTTTGCCGTATCTTTTA GC | 3' hctAprom(LVA) | p 2 T K 2 - *i h t A* p r o m - m N G ( L V A ) _ *h c t A* p r o m - mKate2 |
| GgcggatcCGatggtgagcaaag gcgaagaagataacatgg | 5' hctAprom(LVA) vector | p 2 T K 2 - e u o p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2 |
| ctagaatctaaGACGCGGCC GCGTCTTAGGAGCTTTTT GC | 3' hctAprom(LVA) vector | p 2 T K 2 - e u o p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2 |
| **p2TK2-*euo*prom-mEos3.2** | | |

| atacaggggtcgaattcggcATGTCCGCAATAAAGCCTGACATGAAGATCAAGCTCAGAATGGAAGGCAACGTCAATGGTCATCATTTTGTCATCGACGGTGACGGTACAGGGAAGCCTTTTGAGGGGAAACAGTCAATGGATTTGGAAGTAAAAGAAGGCGGTCCACTTCCTTTTGCTTTCGACATCTTAACCACAGCGTTCCACTACGGAAATCGCGTGTTTGCAAAGTACCCCGATAACATCCAGGACTATTTCAAACAGTCATTTCCAAAAGGCTACTCCTGGGAGAGATCCCTTACGTTCGAAGACGGAGGCATCTGTAACGCACGCAACGATATTACTATGGAAGGTGATACTTTCTATAACAAGGTGCGTTTCTATGGAACCAACTTCCCTGCCAATGGACCTGTTATGCAAAAAAAAACTTTGAAATGGGAGCCAAGTACTGAAAAAATGTATGTACGCGATGGGGTTCTCACAGGAGATATTGAGATGGCACTCTTATTAGAAGGCAACGCTCACTACCGCTGTGATTTCAGAACTACATATAAAGCCAAAGA | mEos3.2 gBlock | |

| | | |
|---|---|---|
| TAGGGATGACATGTGATT CGCGTAGGAAAAGAGG AGGGAGACC | 5' euoprom-mEos3.2 vector | **p2TK2-euoprom-Clover** |
| gccgaattcgacccctgtatcttgttgt aagcattcc | 3' euoprom-mEos3.2 vector | **p2TK2-euoprom-Clover** |
| **p2TK2-*hctA*prom-mEos3.2_*hctB*prom-mKate2** | | |
| GgaattcggcATGTCCGCAAT AAAGCCTGACATGAAGAT CAAGCTCAGAATGG | 5' mEos3.2 | **p2TK2-*euo*prom-mEos3.2** |
| ATCCGTCCTAACGGCGC GCATTATCAGGGAGCCCG GAGTGC | 3' mEos3.2 | **p2TK2-*euo*prom-mEos3.2** |
| TGCGCGCCGTTAGGACG GATCCCTTGTACAATCAAT TTACCGATTAAATAGTCTC | 5' mEos3.2 vector | **p2TK2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2** |
| TTGCGGACATgccgaattcCA TTTTTTTTGCCGTATCTTT TAGCGCCATg | 3' mEos3.2 vector | **p2TK2-*hctA*prom-mNG(LVA)_*hctB*prom-mKate2** |
| **p2TK2-E-*ftsI*3XFLAG** | | |
| TAACAACAAGATGAATCA CCGTAGACAATTAACTCT GATCGTTGTTGGGG | 5' FtsI (E-FLAG) | **L2 Genomic** |
| TGTAGTCcatTTTGCGATT CCATTCCTCATATAGCAG CTTTAATTGAGAAACTTCT TCAC | 3' FtsI (E-FLAG) | **L2 Genomic** |

| | | |
|---|---|---|
| GAATCGCAAAatgGACTAC AAAGACCATGACGGTGAT TATAAAGATCATGACATCG | 5' E-FLAG (FtsI) vector | **p2TK2-T5-E-Clover3XFLAG** |
| GGTGATTCATCTTGTTGTT ACCTCCTTAGCAGGGTG CTGCCAAGG | 3' E-FLAG (FtsI) vector | **p2TK2-T5-E-Clover3XFLAG** |
| **p 2 T K 2 - E - ftsI3XFLAG_*euo*prom- mNG(LVA)_*hctB*prom- mKate2** | | |
| CCGTAAAAAATgttaaaaacta accattttttattaaaagttttttcattctcctt gtcg | 5' *hctB*mKt2 (E-FLAG) | **p 2 T K 2 - *e u o* p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2** |
| AACGCGGCCACTAAGCTA CTAAAGCGTAGTTTTCGT CGTTTGCTGCAGG | 3' *euo*mNG(LVA) (E-FLAG) | **p 2 T K 2 - *e u o* p r o m - m N G ( L V A ) _ *h c t B* p r o m - mKate2** |
| gttttttaacATTTTTTACGGTT CCTGGCCTTTTGCTGGC CTTTTGC | 5' *hctB*mKt2 (E-FLAG) vector | **p2TK2-E-ftsI3xFLAG** |
| AGTAGCTTAGTGGCCGC GTTGCTGGCGTTTTTCC | 3' LVA (E-FLAG) vector | **p2TK2-E-ftsI3xFLAG** |
| **p 2 T K 2 - E - ftsI3XFLAG_*hctA*prom- m E o s 3 . 2 _ *h c t B* p r o m - mKate2** | | |
| CCGTAAAAAATgttaaaaacta accattttttattaaaagttttttcattctcctt gtcg | 5' *hctB*mKt2 (E-FLAG) | **p 2 T K 2 - *h c t A* p r o m - mEos3.2_*hctB*prom-mKate2** |

| | | |
|---|---|---|
| AACGCGGCCACTAACGG CGCGCATTATCAGGGAGC CCGG | 3' mEos3.2 (E-FLAG) | **p 2 T K 2 - *h c t A* p r o m - mEos3.2_*hctB*prom-mKate2** |
| gtttttaacATTTTTTACGGTT CCTGGCCTTTTGCTGGC CTTTTGC | 5' *hctB*mKt2 (E-FLAG) vector | **p2TK2-E-ftsI3xFLAG** |
| ATAATGCGCGCCGTTAGT GGCCGCGTTGCTGGCGT TTTTCC | 3' mEos3.2 (E-FLAG) vector | **p2TK2-E-ftsI3xFLAG** |
| **ddPCR** | | |
| TGGGAAACTTAAGTCCGC TC | 5' *copN* primer | **C t r - L 2 - e u o p r o m - m N G ( L V A ) _ h c t B p r o m - m K a t e 2 o r C t r - L 2 - E - f t s I 3 X F L A G _ e u o p r o m - m N G ( L V A ) _ h c t B p r o m - mKate2** |
| TAAGGAGCGAAGCGATG AAG | 3' *copN* primer | **C t r - L 2 - e u o p r o m - m N G ( L V A ) _ h c t B p r o m - m K a t e 2 o r C t r - L 2 - E - f t s I 3 X F L A G _ e u o p r o m - m N G ( L V A ) _ h c t B p r o m - mKate2** |
| CCTCAGGCGATTGTTGGA GGACGCAATGT | *copN* probe | **C t r - L 2 - e u o p r o m - m N G ( L V A ) _ h c t B p r o m - m K a t e 2 o r C t r - L 2 - E - f t s I 3 X F L A G _ e u o p r o m - m N G ( L V A ) _ h c t B p r o m - mKate2** |

CHAPTER FOUR: LIVE-CELL FORWARD GENETIC APPROACH TO IDENTIFY
AND ISOLATE DEVELOPMENTAL MUTANTS IN CHLAMYDIA TRACHOMATIS*

Travis J. Chiarelli[1], Nicole A. Grieshaber[1], Scott S. Grieshaber[1]

[1]Department of Biological Sciences, University of Idaho, Moscow Idaho, USA

**Abstract**

The intracellular bacterial pathogen *Chlamydia trachomatis* undergoes a developmental cycle consisting of two morphologically discrete developmental forms. The non-replicative elementary body (EB) initiates infection of the host. Once inside, the EB differentiates into the reticulate body (RB). The RB then undergoes multiple rounds of replication, before differentiating back to the infectious EB form. This cycle is essential for chlamydial survival as failure to switch between cell types prevents either host invasion or replication.

Limitations in genetic techniques due to the obligate intracellular nature of *Chlamydia* have hampered identification of the molecular mechanisms involved in the cell-type development. We designed a novel dual promoter-reporter plasmid system that, in conjunction with live-cell microscopy, allows for the visualization of cell type switching in real time. To identify genes involved in the regulation of cell-type development, the live-cell promoter-reporter system was leveraged for the development of a forward genetic approach by combining chemical mutagenesis of the dual reporter strain, imaging and tracking of *Chlamydia* with altered developmental kinetics, followed by clonal isolation of mutants. This forward genetic workflow is a flexible tool that can be modified for directed interrogation into a wide range of genetic pathways.

**Introduction**

*Chlamydia trachomatis* (*Ctr*) is an obligate intracellular pathogen that progresses through a biphasic developmental cycle that is essential for its survival and proliferation [1]. This cycle consists of two developmental forms, the elementary body (EB) and the reticulate body (RB). The EB is replication incompetent but mediates cell invasion through effector induced endocytosis [2]. Once in the host, the EB matures to the replicative RB. The RB carries out multiple rounds of replication prior to converting back to the EB in order to initiate subsequent rounds of infection.

The limited array of genetic tools has restricted most of the chlamydial research to biochemical studies or the use of surrogate systems. As a consequence, elucidation of gene regulation and control of the developmental cycle has been difficult [3, 4]. One of the more important challenges in the chlamydial field is the high resolution temporal tracking of the chlamydial developmental cycle and the identification of the proteins involved in its regulation. Gene expression during the chlamydial developmental cycle has traditionally been performed by destructive "end point" methods including RNAseq, qPCR, and fixed cell microscopy [5, 6]. Although these methods have provided invaluable information, the techniques employed are laborious and have low temporal resolution [5, 6].

Within the last decade, genetic manipulation of *Ctr* has progressed with the introduction of plasmid transformation and methods for mutagenesis [7, 8, 9]. For this study, a plasmid-based system was developed to monitor chlamydial development in individual inclusions in real time over the course of an infection. A chlamydial transformant was created that expressed both an RB and EB cell-type specific promoter-reporter. The RB specific reporter was constructed by fusing the promoter of the early RB gene *euo* upstream of the fluorescent protein Clover. EUO is a transcriptional regulator that represses a subset of late EB associated genes [10]. The promoter of *hctB*, which encodes a histone-like protein involved in EB nucleoid condensation, was cloned directly upstream of mKate2 (RFP) to create the EB specific reporter [11]. The backbone for *hctB*prom-mKate2/*euo*prom-Clover was p2TK2SW2[7]. The *hctB* and *euo* promoters were amplified from *Ctr*-L2 genomic DNA. Each promoter sequence consisted of ~100 base pairs upstream of the predicted

transcription start site for the specified chlamydial gene plus the first 30 nucleotide (10 amino acids) of the respective ORF. The fluorescent FP variants were commercially obtained as *Ctr* codon optimized gene blocks and cloned in frame with the first 30 nucleotide of each chlamydial gene and promoter. The incD terminator was cloned directly downstream of mKate2. The second promoter-reporter was inserted downstream of the incD terminator. The ampicillin resistance gene (bla) in p2TK2SW2 was replaced with the *aadA* gene (Spectinomycin resistance) from pBam4. This resulted in the final construct p2TK2-*hctB*prom-mKate2/ *euo*prom-Clover (**Figure 4.1A**) that was transformed into *Ctr*-L2 [7]. This RB/EB reporter strain allowed for the observation of the developmental cycle within single inclusions using livecell microscopy (**Figure 4.1B,C**).

Employing our promoter-reporter construct in combination with chemical mutagenesis, a protocol was devised to track and isolate individual clones that exhibited developmental abnormalities from mutagenized populations of *Ctr* serovar L2. This protocol allows for the direct monitoring of individual chlamydial inclusions, tracking of the gene expression profiles over time, identifying chlamydial clones that express an altered developmental gene expression pattern, and clonal isolation of *Chlamydia* from individual inclusions.

Although this protocol has been created specifically for the identification of genes involved in chlamydial development, it could be easily adapted to interrogate any number of chlamydial genetic pathways.

**Protocol**

All Python scripts used in this protocol are available on Github https://github.com/ SGrasshopper/Live-cell-data-processing

## 1. Mutagenize Reporter *Chlamydia*

NOTE: *Ctr*-L2-*hctB*prom-mKate2/*euo*prom-Clover EBs were directly mutagenized using ethyl methanesulfonate (EMS) in the axenic media CIP-1 as this media supports EB metabolism and maintenance of EB infectivity [12].

1. Thaw a chlamydial stock on ice containing ~3 x $10^7$ EBs transformed with the p2TK2-*hctB*prom-mKate2/*euo*prom-Clover reporter plasmid and pellet at >14,000 x g for 30 min at 4 °C.

   NOTE: *Chlamydia* organisms used for these experiments were 30% renografin density purified and frozen at −80 °C in 1x sucrose-phosphate-glutamate buffer (SPG).

2. Discard the supernatant and resuspend the EB pellet in 100 μL of CIP-1 buffer with sonication on ice at 10% power for 10 s. Divide the 100 μL of EB suspension into two 50 μL aliquots for mutagenized and mock treated samples.

3. Prepare 20 mg/mL of EMS-CIP-1 solution in a separate 1.5 mL microcentrifuge tube. To do so, add 6.8 μL of EMS in 375 μL total volume.

4. Add 50 μL of the EMS-CIP-1 solution into one of the chlamydial aliquots for mutagenesis and 50 μL of CIP-1 only to the other chlamydial aliquot for mock mutagenesis.

NOTE: Final EMS concentration is 10 mg/mL. The chlamydial titer, EMS concentration, and the time of exposure used in this protocol lead to approximately a 60-80% reduction in infectious progeny. This level of reduction corresponds to ~5-20 DNA lesions per chlamydial genome [8].

5. Incubate for 20 min at room temperature. The mutagenized EBs will be used directly to infect monolayers in section 2.

CAUTION: EMS is a known carcinogen. All equipment and materials that come in contact with EMS must be soaked in 1 M NaOH for 24 h before disposal, gloves should be used at all times during the protocol and cleanup of EMS materials.

## 2. Imaging of mutant *Ctr*

### 1. Host cell culture for imaging and isolation of mutagenized *Ctr*

1. Seed a 6 well glass bottom plate with 6 x $10^5$ Cos-7 cells (ATCC) per well in 2 mL of complete media (RPMI-1640 supplemented with 10% fetal bovine serum and 10 mg/mL gentamicin). Use this glass bottom plate for imaging of mutagenized *Ctr*.

2. Seed a 24 well polystyrene plate with 1 x $10^5$ Cos-7 cells (ATCC) per well in 1 mL complete media. Use this polystyrene plate for reinfection of isolated *Chlamydia* of interest.

3. Incubate both the plates at 5% $CO_2$, 37 °C for approximately 18 h. Once cells reach confluency, replace media with complete media supplemented with 1 µg/mL of cycloheximide and incubate overnight.

### 2. Infecting the host cell culture with mutagenized *Ctr*

**1.** Infect 5 wells of the glass bottom plate with ~6 x 10$^5$ of mutagenized EBs in 1.5 mL/well ice cold HBSS. This will result in the MOI of ~ 0.3 as ~70% mortality rate is expected due to mutagenesis.

**2.** Infect the remaining well with ~2 x 10$^5$ mock mutagenized EBs in 1.5 mL/well ice cold HBSS. Without mutagenesis, expect less mortality, thus one-third of the inoculum is used to achieve the MOI of ~0.3. NOTE: MOI of ~0.3 ensures that host cells are infected by a single EB and allows for enough separation between infected cells for clonal isolation.

**3.** Incubate the plate for 15 min, with rocking, at 37 °C.

**4.** Wash the infected host cells with prewarmed (37 °C) HBSS containing 1 mg/mL heparin followed immediately by an HBSS rinse. Repeat heparin wash, immediately rinsing 2x with HBSS to ensure the heparin is removed.

NOTE: Heparin inhibits and can reverse the early electrostatic interactions between the host cell and EBs [13] . The heparin washes remove EBs that have yet to enter the host cells, synchronizing the infection. When washing cells do so gently to prevent dislodging the cells from the surface of the wells. HBSS and heparin solutions contain residual EMS and should be placed in a beaker containing 1 M NaOH for 24 h before disposal.

**5.** Replace HBSS with 4 mL/well of prewarmed (37 °C) imaging media (complete media, 1 µg/mL cycloheximide, 20 mM HEPES, and no phenol red).

**6.** Fill the interwell spaces with prewarmed (37 °C) deionized $H_2O$ to aid in the temperature control and reduce evaporation. Incubate the plate at 37 °C incubator with 5% $CO_2$ for 10 h.

**3. Microscope set up and imaging**—NOTE: Multicolor multiposition automated live-cell fluorescent imaging is used to collect time-lapse images to identify chlamydial mutants that differ in the developmental gene expression dynamics. This protocol utilizes the open source µManager software package for automated microscope control[14].

**1.** Begin the microscope setup 10 h post infection. Set the microscope stage incubator to 5% $CO_2$, 37 °C. Place the infected 6 well glass bottom plate into the stage incubator and insert the sample thermistor into the interwell $H_2O$.

**2.** Calibrate the XY stage using the High Content Screening (**HCS**) plugin. Click **Plugins ⎮ Acquisition Tools ⎮ HCS Site Generator** in the µManager microscope control software (**JoVE61365_screenfile1, JoVE61365_screenfile2**).

**3.** Select the 6-well plate template and generate an imaging position list consisting of 12 fields of view (FOV) per well within the **HCS** plugin (**JoVE61365_screenfile3, JoVE61365_screenfile4**). Open the **Stage Position List** and manually focus and set the initial Z position for each FOV using the Stage Control plugin (**JoVE61365_screenfile5, JoVE61365_screenfile6**). Adjust the XY coordinates of any FOV that has missing cells or does not contain a uniform monolayer. NOTE: Due to the time it takes for each image to be captured, a maximum number of 72 FOV can be taken per 30 min interval.

**4.** Use a 20x objective lens for imaging. This magnification allows ~8 inclusions to be imaged per FOV while still providing the desired resolution.

**5.** Save the positions list as this will be used to locate the inclusions of interest after data analysis (**JoVE61365_screenfile7**).

**6.** Use the **Auto Focus** option in the imaging software, to set the focus for automated imaging (**JoVE61365_screenfile8**).

**7.** Use the following selections and values to produce the most consistent focus results using image based autofocus in µManager. In the Autofocus properties window select **OughtaFocus** from the drop-down menu and use the following settings. OughtaFocus-SearchRange_µm: 350, OughtaFocus-Tolerance_µm: 0.5, OughtaFocusCropFactor: 0.3, OughtaFocus-Exposure: 20, OughtaFocusFFTLowerCutoff(%): 2.5, OughtaFocus-FFTUpperCutoff(%): 14, OughtaFocusShowImages: Yes, OughtaFocus-Maximize: SharpEdges, OughtaFocus-Channel: DIC.

NOTE: Reducing the autofocus imaging window by decreasing the crop factor allows for more consistent auto focusing. Selecting **Yes** for OughtaFocusShowImages allows the user to view the autofocus image.

**8.** Capture the kinetics of the developmental cycle by imaging for 24 h with 30 min time intervals (**JoVE61365_screenfile9**). Imaging between 12-36 HPI ensures that *Chlamydia* completes the developmental cycle but does not lyse the host cell.

**9.** Image the cell monolayers with a 250 ms exposure at 4% and 18% intensity in the GFP and RFP channels, respectively (**JoVE61365_screenfile10**). Detect the Clover (GFP) signal by excitation

at 470 nm with a 514/30 nm bandpass emissions filter. Detect the mKate2 (RFP) signal by excitation at 595 nm and with a 641/75 nm bandpass emissions filter. NOTE: Minimizing the excitation intensity is critical for minimizing fluorophore photobleaching and phototoxicity to *Chlamydia*. The minimum excitation intensity to generate a resolved image with a 200-300 ms exposure should be determined empirically in pilot studies.

10. Capture multiple Z-slices with a range of focus that ends on either side of the infocus slice. In this experiment, at 20x magnification, this was achieved with 4 slices at 10 μm steps (**JoVE61365_screenfile11**).

11. Select **Relative Z** for imaging multiple slices in the acquisition window. The relative Z option uses the Z plane location saved in the imaging position list as the starting point for the next time interval. Input the appropriate Z-offset values for fluorescence imaging channels (**JoVE61365_screenfile12**). NOTE: The image based focusing system is imperfect and over a 24 h imaging period this leads to focus drift. It was found that by capturing 3-4 Z focal planes for each time point an in-focus image was maintained. Z-offset is needed to correct for the differences in focal planes between the fluorescent image channels and the DIC channel. Empiric determination of this Z-offset will be needed.

12. Save images by selecting the root directory and naming the experiment. Use the μManager **image stack file** option to save images as tiff stack files (**JoVE61365_screenfile13**).

13. Record the experimental details in the **Acquisitions Comments** box in the Multi-D Acquisition window (**JoVE61365_screenfile13**). i.e., Well A1: Untreated control. Well A2-3 and B1-3: EMS Mutants. Imaging: 12-36HPI. Start the image acquisition at 12 HPI. NOTE: If using μManager, leave the program, experimental setup, and microscope hardware running after the

experiment is complete. The imaging sites will be revisited for inclusion isolation once analysis of the mutagenized population is completed.

## 3. Identify and isolate mutagenized *Chlamydia* with altered developmental phenotypes

### 1. Creating an in-focus image stack

1. Extract the most in-focus image from the Z-stacks using the saved image data which contains 4 Z slices per time point. Use the kurtosis measurement option (**Analyze | Measure**) in ImageJ/FIJI to automatically identify the most in focus Z slice (highest kurtosis score) and create a new image stack with just these infocus images. A Python script is included as a supplementary file (**Reduce_Z_kertosis_2ch_JOVE.py**) to automate this process.

### 2. Quantify fluorescence expression in individual inclusions—NOTE: To quantify the expression kinetics of the two reporters use the open source image analysis application ImageJ/FIJI and the plugin Trackmate [15]. Trackmate identifies 'spots' (corresponding to inclusions in this case) and follows them through a time-lapse image stack recording the X,Y location and signal intensity for each inclusion over time (**JoVE61365_screenfile14**). This information is saved as a CSV file and will be imported into a custom Python notebook for analysis.

1. Open the in-focus Z-reduced image stack in ImageJ/FIJI using Bio-formats Importer by clicking **Plugins | Bio-Formats | Bio-formats Importer**. Select Hyperstack and Composite (**JoVE61365_screenfile15, JoVE61365_screenfile16**).

2. Subtract the image background by clicking on **Process �ő Subtract Background** using a Rolling ball radius of 50.0 pixels and enhance the image contrast to 0.3% for Saturated pixels (**Process ⏽ Enhance Contrast**). Multiply the image values by 10.0 (**Process ⏽ Math ⏽ Multiply**) (**JoVE61365_screenfile17 - JoVE61365_screenfile22**).

3. In Trackmate (**Plugins ⏽ Tracking ⏽ Trackmate**), select an estimated blob diameter of 48 pixels (empirically determined based on the size of the inclusion at the end of imaging). Produce non-fragmented inclusion tracks by selecting a **Linking max distance** and **Gap-closing max distance** of 8.0 pixels and **Gap-closing max frame gap** of 1 (**JoVE61365_screenfile23 - JoVE61365_screenfile25**).

NOTE: To improve Trackmate's ability to identify and track inclusions over the entire cycle a separate image channel is created by adding the *euo*prom and *hctB*prom channels together using the image math function in ImageJ/FIJI. This channel is then used by Trackmate to identify and follow inclusions over time. The fluorescent values of the *euo*prom and *hctB*prom channels are then recorded in channels 2 and 3.

4. Record tracks that meet a minimum continuous duration: Duration of track: 20 (**JoVE61365_screenfile26**). Analyze the tracks and save the **Spots in track** statistics as a CSV file (**JoVE61365_screenfile27**). NOTE: This process has been automated using a custom Python script that is provided in the supplemental data (**TrackMate_Zreduced_JOVE.py**).

**3. Identify inclusion tracks with altered developmental profiles**—NOTE: To identify inclusions containing *Chlamydia* with altered developmental profiles, each inclusion track was visualized using Python notebook. These

visualizations allow for the identification of inclusions with kinetic gene expression profiles that differed from the mock-treated population. The Python notebook used for identification of inclusions with altered developmental programming is provided in the supplemental data (EMS_ScreenMarkdown).

1.  Import the inclusion track data from the **Spots in tracks statistics** CSV files into Pandas data frame using the **Import** cell in the EMS_Screen-Markdown Python Notebook.

2.  Baseline correct each track by subtracting the minimum value of each track from the rest of the track values using the **Baseline Subtract** cells. Save the resulting values as a pickle file using the **Save as Pickle** cell. This will permanently save the channel values after baseline subtraction for later retrieval. NOTE: Baseline subtraction sets the starting fluorescent intensity of every inclusion to zero.

3.  Eliminate traces from inclusions near the edges of the FOV using the **Filter Edges** cell as their fluorescent profiles may not be fully captured; these traces may produce false positive developmental profiles.

4.  Calibrate the frame (**totalFrames**) values from the image slices to time (startTime, interval) values with the **Time-lapse Calibration** cell using the experimental start time and imaging time interval. Exclude partial inclusion reads by filtering out traces that do not extend over the last 20 h of the experiment (16-36 HPI) using the **Track Duration Filter** cell.

5.  Separate tracks into individual data frames by experimental condition with the **Assign Treatment** cell. Filter for inclusions that exhibit sufficient growth using the **Filter for Growth** cell. In the Filter for Growth cell, empirically set the fluorescence intensity threshold for the

*euo*prom channel by changing the values within the last two lines of code. This will filter out *Chlamydia* that did not grow. NOTE: Be cautious when setting threshold filters, if the filter is too low the resulting scatter plots will be noisy, yet if filtering is set too high important mutants may be eliminated.

6. Calculate the percent mortality caused by EMS by dividing the number of mutagenized tracks/well by the number of mock-treated tracks/well. Multiply the number of mock-treated tracks by the initial dilution factor of 3 to calculate the number of mock-treated tracks/well. Use the **Count Inclusion Tracks** and **Calculate Percent Mortality** cells to perform this task.

7. Calculate the time to half-maximal expression for both early and late reporters for each track using the **Calculate** cell. These values will be used to compare mutagenized and mock populations to identify developmental mutants.

8. Within the **Half-Max Plot** cell use the bokeh plotting package to visualize the time to half-max expression of each promoter, graphing the *euo*prom time to half-maximal expression against that of *hctB*prom. Identify inclusions from the mutant population that fall outside the mock-treated scatter cloud using the bokeh interactive track ID explorer. Make note of the FOV and XY coordinates of the inclusions of interest (**Figure 2.2**).

   NOTE: Pick candidate inclusions that visually fall outside of the control cloud as verification and statistical evaluation of each clone will be performed subsequently.

9.   Within the **Animated Plot** cell visualize changes in promoter expression kinetics dynamically through time by graphing the expression intensities of *euo*prom against *hctB*prom using the plotting tool Plotly[16]. The scatter function of Plotly is used to animate the gene expression over time (**Video 4.1**).

10. Visualize a snapshot from the Plotly animated graph in the **Inclusion Locator** cell, plotting *euo*prom and *hctB*prom expression at a specific time point (i.e. 28 HPI) using the bokeh package (**Figure 4.3**). Identify inclusions from the mutant population as described in step 3.3.8.

NOTE: This analysis needs to be performed quickly (<4 h) as the inclusions are still expanding and will start to lyse host cells ~48 h after infection.

**4. Isolate developmental mutants from inclusions of interest**—NOTE: To isolate *Chlamydia* from the inclusions that were determined to display altered gene regulation a micromanipulator with capillary needles was employed. The Well ID, FOV and X,Y coordinates of inclusions of interest were determined using the data visualization in section 3.3.

1. Prepare capillary needles by holding the center of the capillary tube in a flame and pulling both ends of the capillary tube until it has separated. Create an opening in the pulled capillary needle by breaking the pulled tip on a microscope slide. To break the needle, place the closed tip on the frosted portion of the microscope slide at an angle and apply pressure.

NOTE: Capillary tube specifications were 1.0 mm O.D., 0.5 mm I.D.

**2.** Check that the needle opening is approximately the size of an inclusion under a microscope, 20x objective.

**3.** Prepull ~25-30 capillary needles for isolation of candidate inclusions (one needle per inclusion).

**4.** Fill the microinjector with mineral oil ensuring that no air bubbles are present.

**5.** Attach a glass capillary needle to the microinjector and expel oil to the tip of the needle, expunging any air bubbles. Place the capillary needle in complete media and draw media up halfway. Filling the capillary needle with media prevents oil contamination in the well.

**6.** Using the saved position list in μManager from step 2.3.5, migrate to the well and FOV of an inclusion of interest identified in the Python visualization notebook.

**7.** Use the joystick of the micromanipulator to localize the capillary needle to the XY coordinates of the inclusion of interest.

**8.** Use the 595 nm excitation channel to visualize EBs for extraction and the phase/DIC white light channel for needle visualization. Maneuver the capillary needle to the inclusion, rupture the inclusion and then draw the EBs into the capillary needle using the microinjector.

**9.** Expel the EBs from the capillary needle into a single well of the prepared 24 well polystyrene plate prepared in step 2.1.2. Remove the capillary needle and replace with a fresh capillary needle for next inclusion extraction. Repeat section 3.4 for all candidate inclusions.

NOTE: For expansion and to ensure a high enough titer for re-imaging, incubate mutant isolates in a 5% $CO_2$, 37 °C incubator until the majority of the host cells are infected (~1 week). Wells should be monitored closely as different isolates may exhibit different growth rates.

## 5. Harvest mutant isolates

1. On ice, disrupt the infected monolayer by scraping with a 1 mL micropipette tip. Transfer the media, cell debris, and released *Chlamydia* into a 1.5 mL microcentrifuge tube.

2. Pellet *Chlamydia* by centrifugation for 30 min at 4 °C, >14,000 x g. Remove the supernatant and resuspend pellet in 75 μL of ice cold 1x SPG. Aliquot into three 1.5 mL screw-cap microcentrifuge tubes. Store at −80 °C.

## 4. Verification of mutant isolate phenotypes

### 1. Host cell culture for imaging mutagenized isolates

1. Seed a 96 well glass bottom plate with 1.6 x $10^4$ Cos-7 cells (ATCC) per well in 100 μL of complete media. Incubate at 5% $CO_2$, 37 °C. Cells should reach confluency in approximately 24 h. After cells are confluent, replace media with complete media supplemented with 1 μg/mL cycloheximide, incubate overnight.

### 2. Infect cells with candidate isolates for phenotypic verification

1. Thaw mutant clones and wildtype *Chlamydia* on ice.

**2.** In the prepared 96 well plate, perform a two-fold serial dilution of mutant isolates, using one column per isolate (11 columns). Start with an initial dilution of 1:20 in 100 μl HBSS. NOTE: Serial dilution of *Chlamydia* is performed to ensure mutant samples are imaged at an MOI < 1.

**3.** Infect the remaining (12th) column with wildtype *Chlamydia* at MOI ~0.5. NOTE: Wildtype *Chlamydia* are used as a control for comparison against mutagenized isolates.

**4.** Incubate for 15 min rocking at 37 °C.

**5.** Wash infected host cells with prewarmed (37 °C) HBSS with 1 mg/mL of heparin and HBSS as specified in section 2.2.

**6.** Replace with 200 μL per well of prewarmed (37 °C) imaging media.

**7.** Fill the interwell spaces with prewarmed (37 °C) deionized $H_2O$.

**8.** Incubate at 5% $CO_2$, 37 °C for 10 h.

**3. Microscope setup**—NOTE: Refer to section 2.3 for microscope setup, this section will only contain the required setup modifications.

**1.** Select the **96 well plate template** from the **HCS** plugin.

**2.** Empirically determine wells corresponding to an MOI < 1 for each mutant isolate. Clover expression under control of the *euo* promoter is observable at ~10 HPI making early visualization of inclusions possible (**Figure 2.1B**).

**3.** Select three wells per mutant isolate that correspond to an MOI < 1 and generate an imaging position list consisting of two FOV per well.

NOTE: Only 72 images can be taken per time interval due to hardware constraints, this equates to three dilutions (wells) per strain using two imaging sites per well if 12 samples are imaged.

**4.** Record the developmental cycle of each mutant isolate for 36 h at 30 min time intervals starting at 12 HPI.

**5.** Record the experimental details in the **Acquisitions Comments** box. i. e., Well ABC1: wildtype control. Well ABC2: Mutant strain 1, ABC3: Mutant strain 2, etc… Imaging: 12-48 HPI.

**6.** Start the image acquisition at 12 HPI.

## 5. Data analysis for isolate verification

### 1. Create in-focus image stacks and quantify fluorescence expression in individual inclusions

**1.** Generate fluorescent intensity traces for each inclusion as specified in section 3.1 - 3.2.

**2. Verify *Ctr* mutagenized isolates—**NOTE: To verify the altered developmental profiles of mutant isolates, their expression profiles are compared to the wildtype expression profile using Python notebook. The Python notebook used for verification of mutant clones with altered developmental programming is provided in the supplemental data (**clone_check-Markdown**).

1. Import and filter the inclusion trace data in the clone_check-Markdown Python notebook as done in section 3.3.

2. Calculate the mean and standard deviation (STD) from the traces of each isolate and wildtype control population using the **Calculate Mean & STD** cell.

3. With the **Graph Iso vs WT** cell plot the mean and standard error of the mean (SEM) of each mutant clone against the wildtype control to determine if the mutant expression kinetics are divergent from the wildtype sample (**Figure 4.4**).

4. Determine if the isolated mutant population is clonal by plotting the mutant traces and comparing them to wildtype inclusion traces using a scatter plot as done in section 3.3 (steps 3.3.7-3.3.10) (**Figure 4.2, Figure 4.3**). If the isolate is a mixed population the plot will show one population overlaying with wildtype and a second distinct population outside of the wildtype scatter cloud. If the population looks mixed the mutant can be re-isolated using the original procedure described in section 3.4.

   NOTE: To determine if the developmental profile of an isolate is statistically different from wildtype the curves for each isolate should be compared to wildtype using ANOVA.

**Representative Results**

Direct EMS mutagenesis of our promoter-reporter chlamydial strain resulted in an ~75% reduction in infectivity. Using the described live-cell imaging protocol, ~600 inclusions were imaged and tracked over a 24 h period. The fluorescent expression kinetics of both reporters in each inclusion was visualized using custom Python notebook scripts.

Two visualization approaches were implemented to identify candidate mutagenized *Chlamydia* for isolation. The first methodology (step 3.3.8) visualizes the time to halfmaximal expression of *euo* and *hctB* promoters from individual chlamydial isolates in an interactive scatter plot (**Figure 4.2**). Inclusions were identified for isolation if they fell outside the mock-treated scatter cloud. Candidate clones were picked that visually fell outside of the control cloud. Verification of each clone was performed subsequently. Clones A3-6-67 and B3-8-58 were selected for isolation as they produced shorter times to half-maximal expression from the *euo* promoter and longer times for *hctB* (**Figure 4.2**).

The second visualization method for identifying inclusions with altered kinetics (steps 3.3.9-10) identifies individual inclusions based on visualization of dynamic gene expression from the two promoters (**Video 4.1**). Again, candidate clones with dynamic inclusion expression patterns that were noticeably distinct from control inclusions were picked. B3-6-62 was chosen due to increased fluorescent accumulation from the *euo* promoter between 23 and 29 HPI (**Video 4.1**). A snapshot of the animated graph was taken to identify the location of the inclusions of interest (**Figure 4.3**).

Using the two visualization methods, a total of 24 inclusions were identified for isolation. Of the 24 total isolates, 10 showed differential kinetics upon retesting. These isolates fell into three phenotypic categories; 8 isolates exhibited decreased *euo*prom expression at ~24 HPI, corresponding to the time of RB-EB conversion, as demonstrated by the clone A3-6-67 (**Figure 4.4A**). The remaining two clones displayed unique phenotypic profiles, the B3-8-58 isolate also exhibited decreased *euo*prom expression at ~24 HPI, yet an overall increase in *hctB*prom expression (**Figure 4.4B**), whereas B3-6-62 expressed increased levels of fluorescence from the *euo* promoter followed by a sudden loss of expression in both promoters (**Figure 4.4C**). Analysis of the live-cell micrographs for mutant B3-6-62 revealed that host cell lysis occurred in cells infected with this mutant much earlier than in wildtype infected cells (**Video 4.2**).

**Discussion**

Dissecting the mechanisms that control the chlamydial developmental cycle has been hindered by the limitations of the currently available genetic tools. Employing our promoter-reporter *Chlamydia* in conjunction with live-cell automated microscopy, a system was built which enables monitoring of cell-type development in individual inclusions over a 24 h period. This system, in combination with chemical mutagenesis and direct inclusion isolation has established a method to rapidly and clonally select *Chlamydia* expressing altered developmental profiles (**Figure 4.5**).

Chlamydial EBs are metabolically active outside the host when provided with intracellular ionic conditions and an energy source [5, 12]. This EB axenic metabolism was leveraged to mutagenize purified EBs outside of host cells. In this protocol, metabolizing EBs were directly mutagenized with EMS. It was observed that EMS treatment effectively reduced EB viability and generated EBs that produced variable developmental kinetics as expected.

It is estimated that the described EMS mutagenesis protocol generates ~5-20 DNA changes/EB. The live-cell microscopy workflow described is capable of imaging ~8 inclusions per field of view (FOV) and 72 FOVs every in a 30 min interval. Therefore, it is estimated that the effects of ~3000-10,000 mutations can be visualized per run. Multiple runs (3-5) will result in visualization of the effects of 9,000-50,000 mutations. The *Ctr*-L2 genome encodes ~850 genes, suggesting this protocol will result in the visualization of >10 mutations per gene. These estimates indicate that genome coverage, while not complete, should be sufficient.

The strength of this protocol is the ability to track and record the expression kinetics of multiple promoter-reporters at the single inclusion resolution in near real-time. Forward genetics relies on observable phenotypes and clonal isolation. Past methods for forward genetics in *Chlamydia* relied on static observations and plaquing with agar overlays [8]. With our methodology, dynamic promoter activity is recorded throughout the developmental cycle and then visualized to identify inclusions that contain *Chlamydia* with altered gene expression kinetics. Identifying candidate inclusions using multiple parameters (i.e., the time to half-maximal expression and total fluorescent intensity at a given time point) results in distinct

mutant pools that display different developmental kinetics. These *Chlamydia* are likely to have unique mutations that affect the regulation of separate genetic pathways. The fact that these profiles can be recorded live and visualized after a few hours allows time to locate and isolate the inclusions of interest from the infected monolayer. Although we focused on the gene expression dynamics during development, alternative gene reporters can be used to probe other regulatory pathways.

Depending on the genetic pathways being interrogated, caution should be taken with the addition of cycloheximide to host cells. Although incubation with cycloheximide improves the imaging characteristics of the monolayer by blocking replication of the host cells; this effect is achieved through inhibiting host protein synthesis. Inhibition of de novo host protein synthesis could influence the results of the genetic screen depending on the question asked.

Phototoxicity and photobleaching are major hurdles in longterm time-lapse microscopy. To overcome these issues, the specific characteristics of each fluorescent protein should be considered prior to experimentation. Clover and mKate2 have short maturation times (20-30 m), are photostable, and exhibit relatively large quantum yields [17, 18]. These qualities allow for the reduction of excitation intensity and exposure time, thus reducing the amount of phototoxicity and photobleaching incurred. The phase/DIC white light channel was employed for autofocusing as this spectrum of light was less phototoxic to *Chlamydia*.

For this protocol, EMS was used as a chemical mutagen. EMS causes G:C to A:T transitions via guanine alkylation [19]. However, this protocol can be expanded to include alternative mutagens that can induce other kinds of genomic mutations. For instance, acridines are a class of DNA intercalating compounds which induce indels, increasing the chance of frame shifts and therefore null mutations [20].

With advances in chlamydial transformation techniques, mutated genes that are associated with phenotypic complementation groups can be knocked out via insertional gene disruption and genetic complementation for verification of genotype-phenotype linkage [9]. Recovering mutants that block RB to EB development could be problematic as mutations of interest may produce *Chlamydia* that cannot reinfect

host cells. This technique can be modified to identify developmental genes by statistical associations (GWAS). The genomes of *Chlamydia* from isolated inclusions can be directly sequenced without expansion and verification. The high throughput nature of this technique would make statistical associations possible. Again, verification of these associations can be tested through gene disruption and complementation [9].

**Acknowledgments**

We thank Dr. Anders Omsland at Washington State University for supplying the CIP-1 axenic media. This work was supported by NIH grant R01AI130072, R21AI135691 and R21AI113617. Additional support was provided by startup funds from the University of Idaho and the Center for Modeling Complex Interactions through their NIH grant P20GM104420.

## References

1.  AbdelRahman Y, Belland R. The chlamydial developmental cycle. FEMS Microbiology Reviews. 29 (5), 949–959 (2005). [PubMed: 16043254]

2.  Clifton DR, Fields KA, Grieshaber SS, Dooley CA, Fischer ER, Mead DJ, Carabeo RA, Hackstadt T. A chlamydial type III translocated protein is tyrosine-phosphorylated at the site of entry and associated with recruitment of actin. Proceedings of the National Academy of Sciences U. S. A 101 (27), 10166–10171 (2004).

3.  Yu HHY, Tan. M σ28 RNA polymerase regulates *hctB*, a late developmental gene in *Chlamydia*. Molecular Microbiology. 50 (2), 577–584 (2003). [PubMed: 14617180]

4.  Koo I, Stephens R. A Developmentally Regulated Two-component Signal Transduction System in *Chlamydia*. Journal of Biological Chemistry. 278 (19), 17314–17319 (2003).

5.  Grieshaber S, Grieshaber N, Yang H, Baxter B, Hackstadt T, Omsland A. Impact of Active Metabolism on Elementary Body Transcript Profile and Infectivity. Journal of Bacteriology. 200 (14), e00065–18 (2018). [PubMed: 29735758]

6.  Belland RJ, Zhong G, Crane DD, Hogan D, Sturdevant D, Sharma J, Beatty WL, Caldwell HD. Genomic transcriptional profiling of the developmental cycle of. Proceedings of the National Academy of Sciences U. S. A 100 (14), 8478–8483 (2003).

7.  Wang Y, Kahane S, Cutcliffe LT, Skilton RJ, Lambden PR, Clarke IN. Development of a transformation system for *Chlamydia trachomatis*: restoration of glycogen biosynthesis by acquisition of a plasmid shuttle vector. PLoS Pathogens. 7 (9), e1002258. (2011). [PubMed: 21966270]

8.  Nguyen B, Valdivia R. Virulence determinants in the obligate intracellular pathogen *Chlamydia trachomatis* revealed by forward genetic approaches. Proceedings of the National Academy of Sciences U. S. A 109 (4), 1263–1268 (2012).

9. Mueller K, Wolf K, Fields K, Maurelli A. Gene Deletion by Fluorescence-Reported Allelic Exchange Mutagenesis in *Chlamydia trachomatis*. MBio. 7 (1), e01817–15 (2016). [PubMed: 26787828]

10. Rosario CJ, Tan M. The early gene product EUO is a transcriptional repressor that selectively regulates promoters of Chlamydia late genes. Mol Microbiol. 84 (6), 1097–1107 (2012). [PubMed: 22624851]

11. Brickman T, Barry C, Hackstadt T. Molecular cloning and expression of *hctB* encoding a strainvariant chlamydial histone-like protein with DNA-binding activity. Journal of Bacteriology. 175 (14), 4274–4281 (1993). [PubMed: 7687246]

12. Omsland A, Sager J, Nair V, Sturdevant D, Hackstadt T. Developmental stage-specific metabolic and transcriptional activity of *Chlamydia trachomatis* in an axenic medium. Proceedings of the National Academy of Sciences U. S. A 109 (48), 19781–19785 (2012).

13. Su H, Raymond L, Rockey DD, Fischer E, Hackstadt T, Caldwell HD. A recombinant *Chlamydia trachomatis* major outer membrane protein binds to heparan sulfate receptors on epithelial cells. Proceedings of the National Academy of Sciences U.S.A 93 (20), 11143–11148 (1996).

14. Edelstein AD, Tsuchida MA, Amodaj N, Pinkard H, Vale RD, Stuurman N. Computer control of microscopes using µManager. Current Protocols in Molecular Biology. Chapter 14–20 (2010).

15. Tinevez JY, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, Bednarek SY, Shorte SL, Eliceiri KW. TrackMate: An open and extensible platform for single-particle tracking. Methods. 115, 80–90 (2017). [PubMed: 27713081]

16. Plotly Technologies Inc. Collaborative data science. Montréal, QC, https://plot.ly. (2015).

17. Lam AJ, St-Pierre F, Gong Y, Marshall JD, Cranfill PJ, Baird MA, McKeown MR, Wiedenmann J, Davidson MW, Schnitzer MJ, Tsien RY, Lin MZ. Improving FRET dynamic range with bright green and red fluorescent proteins. Nature Methods. 9 (10), 1005–1012 (2012). [PubMed: 22961245]

18. Shcherbo D, Murphy CS, Ermakova GV, Solovieva EA, Chepurnykh TV, Shcheglov AS, Verkhusha VV, Pletnev VZ, Hazelwood KL, Roche PM, Lukyanov S, Zaraisky AG, Davidson MW, Chudakov DM. Far-red fluorescent tags for protein imaging in living tissues. Biochemical Journal. 418 (3), 567–574 (2009).

19. Sega G. A review of the genetic effects of ethyl methanesulfonate. Mutation Research. 134 (2-3), 113–142 (1984). [PubMed: 6390190]

20. Ferguson L, Denny W. Frameshift mutagenesis by acridines and other reversibly binding DNA ligands. Mutagenesis. 5 (6), 529–540 (1990). [PubMed: 2263212]

**Figure 4.1**
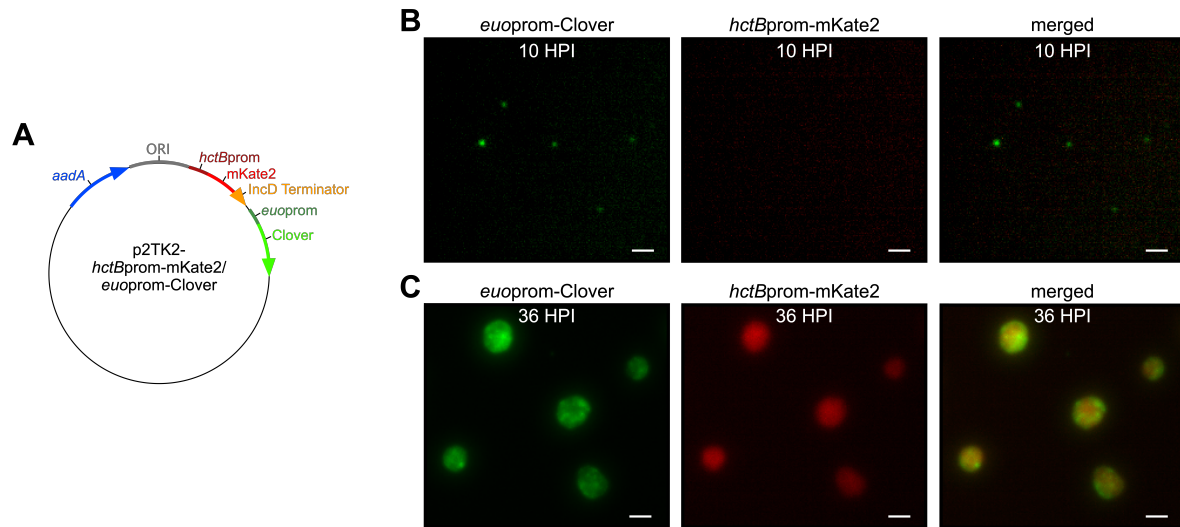


**Figure 4.1: Monitoring cell-type development with *Ctr* promoter-reporters.**

(A) Schematic of the promoter-reporter construct, p2TK2-hctBprom-mKate2/ euopromClover. (B) Live-cell micrograph of euoprom-Clover and hctBprom-mKate2 expression in Ctr at 10 HPI (C) Live-cell micrograph of Ctr expressing euoprom-Clover and hctBprommKate2 at 36 HPI. Scale bar: 20 µm.
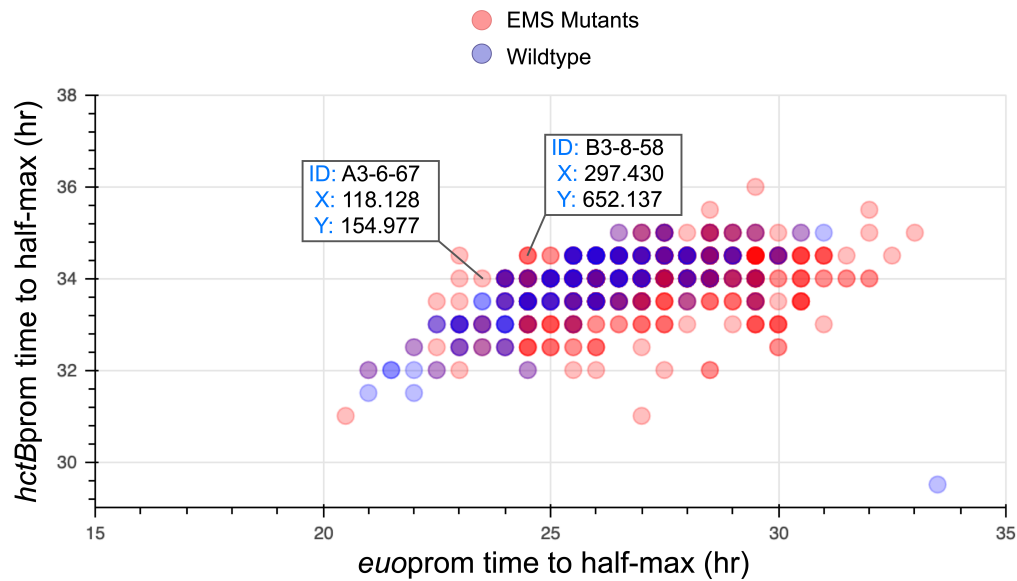
**Figure 4.2**



**Figure 4.2: Identification of representative isolates A3-6-67 and B3-8-58 by visualization of the time to half-maximal expression for each promoter.**

The interactive graph is used to identify mutagenized *Chlamydia* exhibiting expression profiles that differ from the mock-treated control scatter cloud. Each spot on the graph represents a single inclusion. Inclusion spots A3-6-67 and B3-8-58 are highlighted as they fall outside of the mock-treated cloud, both exhibiting shorter time to half-maximal expression of the *euo* promoter in combination with longer time to half-maximal expression of *hctB*. *euo*prom: x-axis, *hctB*prom: y-axis.

**Figure 4.3**
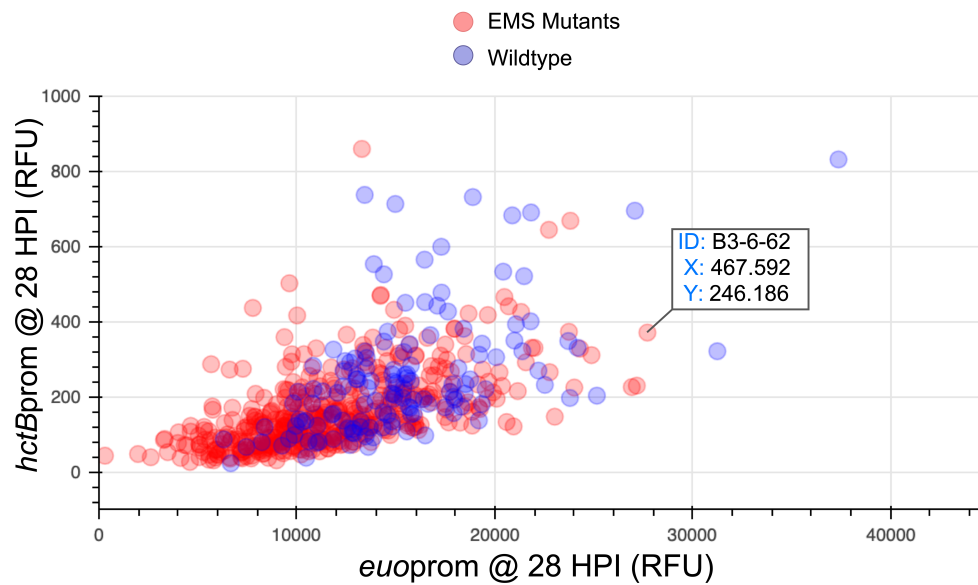


**Figure 4.3: Interactive snapshot for identification of inclusion location.**

The graph presented is a snapshot at 28 HPI from the animated scatter plot (**Video 4.1**) and was used to identify the FOV and XY coordinate location of inclusions of interest. B3-6-62 is shown as it was chosen for isolation from the animated scatter plot.

**Figure 4.4**
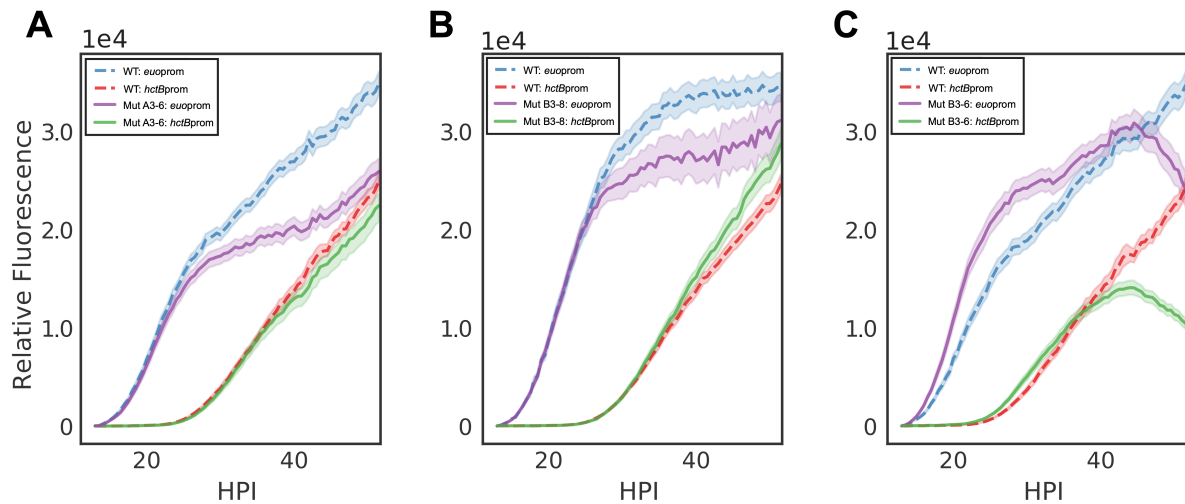


**Figure 4.4: Verification of representative mutant isolates.**

Developmental profiles of mutagenized isolates A3-6, B3-8, and B3-6. (A) The A3-6 mutant exhibits a decrease *euo*prom expression at ~24 HPI. (B) The B3-8 mutant isolate exhibits a decrease *euo*prom expression at ~24 HPI, but an overall increase in *hctB*prom expression. (C) The B3-6 isolate exhibits increased levels of *euo*prom expression followed by a sudden loss of expression in both promoters at ~40 HPI. Each sample is the average of the specified population, n > 25. Cloud represents SEM.

**Figure 4.5**
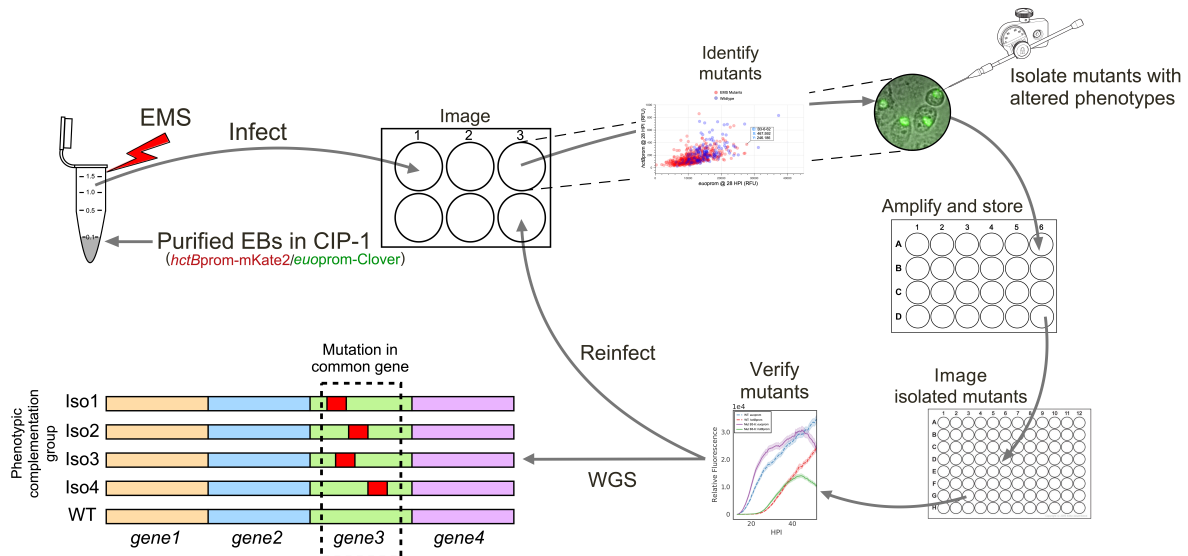


**Figure 4.5: Workflow for directed forward genetic analysis of promoter-reporter *Ctr*.**

*Ctr*-L2-p2TK2-*hctB*prom-mKate2/*euo*prom-Clover EBs were directly mutagenized with EMS in axenic media, CIP-1. Mutagenized EBs were used to infect Cos-7 cell monolayers for imaging and fluorescent expression analysis. *Chlamydia* expressing altered developmental dynamics were identified by visualization in interactive graphs. Inclusions with altered developmental profiles were isolated using a micromanipulator. The phenotypes of the isolates were verified upon reinfection. Mutant isolates are subjected to WGS to identify DNA lesions associated with phenotypes.

## Supplemental Material 4.S1: ReduceZ Kertosis

```python
1  from net.imglib2.img.display.imagej import ImageJFunctions
2  from java.awt.event import TextListener
3  from ij import Menus
4  from ij.gui import GenericDialog
5  from ij.io import OpenDialog
6  from ij.measure import ResultsTable
7  from ij.gui import WaitForUserDialog
8  from ij.plugin import ZProjector
9  from ij.plugin import ChannelSplitter
10 from ij.plugin import RGBStackMerge
11 from ij.plugin import HyperStackConverter
12 import java.util.ArrayList as ArrayList
13 from ij import ImagePlus
14 import csv
15 import os
16 import sys
17 from ij import IJ
18
19 def reduceZ():
20     imp = IJ.getImage()   #get the standardtack
21     title_1 = imp.getTitle()
22     title = title_1.split(' - ')[1]
23     print(title)
24     dimentions = imp.getDimensions()
25     numZ, numChannels, numframes  = dimentions[3], dimentions[2], dimentions[4]
26     print(numChannels)
27
28
29     IJ.run(imp, "Set Measurements...", "kurtosis redirect=None decimal=3")
30
31     kurtTotal = []
32     for time in range(numframes):
33         print(time)
34         time = time+1
35         imp.setPosition(1, 1, time)
36         kurt = []
37         for z in range(numZ):
38             z = z+1
39             imp.setPosition(1, z, time)
40             imp.setRoi(70,40,437,459)
41             IJ.setAutoThreshold(imp, "MaxEntropy dark")
42             IJ.run(imp, "Measure", "")
43             IJ.resetThreshold(imp)
44             rt = ResultsTable()
45             t = rt.getResultsTable()
46             kurt.append(t.getValueAsDouble(23, z-1)) # 23 = kurtosis
47         kurtTotal.append(kurt.index(max(kurt))+1)
48         IJ.run(imp, "Clear Results", "")
```

```python
49      print(kurtTotal)

50

51      IJ.run(imp, "Select All", "")

52

53      imp2 = IJ.createImage("GFP", "16-bit black", dimentions[0], dimentions[1],
…  numframes)
54      imp2 = HyperStackConverter.toHyperStack(imp2, 1, 1, numframes, "Color")
55      print(' ------------')
56      print(numframes)
57      channel = 1
58      i = 0
59      for time in range(numframes):
60          time = time+1
61          imp.setPosition(channel, kurtTotal[i], time)
62          imp.copy()
63          imp2.setPosition(channel, 1, time)
64          imp2.paste()
65          print(time)
66          i=i+1
67      IJ.run(imp2, "Delete Slice", "delete=slice")
68      imp2.show()

69

70      imp4 = IJ.createImage("RFP", "16-bit black", dimentions[0], dimentions[1],
…  numframes)
71      imp4 = HyperStackConverter.toHyperStack(imp4, 1, 1, numframes, "Color")
72      print(' ------------')
73      channel = 2
74      i = 0
75      for time in range(numframes):
76          time = time+1
77          imp.setPosition(channel, kurtTotal[i], time)
78          imp.copy()
79          print(imp.title)
80          imp4.setPosition(channel, 1, time)
81          imp4.paste()
82          i=i+1
83      IJ.run(imp4, "Delete Slice", "delete=slice")
84      imp4.show()

85

86      IJ.selectWindow(title_1)
87      IJ.run("Close")

88

89      imp5 = ImagePlus()
90      IJ.run(imp5, "Merge Channels...", "c1=RFP c2=GFP create")
91      imp5 = IJ.getImage()
92      IJ.run(imp5, "Bio-Formats Exporter",
…  "save=/home/rickettsia/Desktop/data/Clamydial_Image_Analysis/EMS_BMEC_20X_01192020/
…  Zreduced/" + title + ".ome.tif export compression=LZW")
```

## Supplemental Material 4.S2: TrackMate Z-Reduced

```python
1  from fiji.plugin.trackmate.detection import BlockLogDetectorFactory
2  from fiji.plugin.trackmate.detection import LogDetectorFactory
3  from fiji.plugin.trackmate.features.spot import SpotIntensityAnalyzerFactory
4  from fiji.plugin.trackmate.features.spot import SpotContrastAndSNRAnalyzerFactory
5  import fiji.plugin.trackmate.tracking.sparselap.SparseLAPTrackerFactory as
…  SparseLAPTrackerFactory
6  import
…  fiji.plugin.trackmate.extra.spotanalyzer.SpotMultiChannelIntensityAnalyzerFactory
…  as SpotMultiChannelIntensityAnalyzerFactory
7  from fiji.plugin.trackmate.tracking import LAPUtils
8  import fiji.plugin.trackmate.features.FeatureFilter as FeatureFilter
9  import fiji.plugin.trackmate.features.track.TrackDurationAnalyzer as
…  TrackDurationAnalyzer
10  import fiji.plugin.trackmate.features.track.TrackSpotQualityFeatureAnalyzer as
…  TrackSpotQualityFeatureAnalyzer
11  import fiji.plugin.trackmate.SelectionModel as SelectionModel
12  import fiji.plugin.trackmate.visualization.hyperstack.HyperStackDisplayer as
…  HyperStackDisplayer
13  import fiji.plugin.trackmate.Settings as Settings
14  import fiji.plugin.trackmate.Model as Model
15  import fiji.plugin.trackmate.TrackMate as TrackMate
16  import fiji.plugin.trackmate.Spot as Spot
17  from ij.plugin import ChannelSplitter
18  from ij.plugin import ImageCalculator
19  from net.imglib2.img.display.imagej import ImageJFunctions
20  from java.awt.event import TextListener
21  from ij import Menus
22  from ij.gui import GenericDialog
23  from ij.io import OpenDialog
24  from ij.measure import ResultsTable
25  from ij.gui import WaitForUserDialog
26  import java.util.ArrayList as ArrayList
27  import csv
28  import os
29  import sys
30  from ij import IJ
31  from ij import ImagePlus
32
33  def track():
34      imp = IJ.getImage()
35      nChannels = imp.getNChannels()  # Get the number of channels
36      orgtitle = imp.getTitle()
37      IJ.run("Subtract Background...", "rolling=50 sliding stack")
38      IJ.run("Enhance Contrast...", "saturated=0.3")
39      IJ.run("Multiply...", "value=10 stack")
40      IJ.run("Subtract Background...", "rolling=50 sliding stack")
41      IJ.run("Set Scale...", "distance=0")
42
```

```python
43      channels = ChannelSplitter.split(imp)
44      imp_GFP = channels[0]
45      imp_RFP = channels[1]
46      IJ.selectWindow(orgtitle)
47      IJ.run("Close")
48      ic = ImageCalculator()
49      imp_merge = ic.run("Add create stack", imp_GFP, imp_RFP)
50      imp_merge.setTitle("add_channels")
51      imp_merge.show()
52      imp_RFP.show()
53      imp_GFP.show()
54
55      imp5 = ImagePlus()
56      IJ.run(imp5, "Merge Channels...", "c1=[" + imp_merge.title + "] c2=" +
…  imp_GFP.title + ' c3=' + imp_RFP.title + " create")
57      print("c1=[" + imp_merge.title + "] c2=" + imp_GFP.title + ' c3=' +
…  imp_RFP.title + " create")
58      imp5.show()
59      imp5 = IJ.getImage()
60
61      nChannels = imp5.getNChannels()
62      # Setup settings for TrackMate
63      settings = Settings()
64      settings.setFrom(imp5)
65
66      # Spot analyzer: we want the multi-C intensity analyzer.
67      settings.addSpotAnalyzerFactory(SpotMultiChannelIntensityAnalyzerFactory())
68
69      # Spot detector.
70      settings.detectorFactory = LogDetectorFactory()
71      settings.detectorSettings = settings.detectorFactory.getDefaultSettings()
72      settings.detectorSettings['TARGET_CHANNEL'] = 1
73      settings.detectorSettings['RADIUS'] = 24.0
74      settings.detectorSettings['THRESHOLD'] = 0.0
75
76      # Spot tracker.
77      # Configure tracker - We don't want to allow merges or splits
78      settings.trackerFactory = SparseLAPTrackerFactory()
79      settings.trackerSettings = LAPUtils.getDefaultLAPSettingsMap() # almost good
…  enough
80      settings.trackerSettings['ALLOW_TRACK_SPLITTING'] = False
81      settings.trackerSettings['ALLOW_TRACK_MERGING'] = False
82      settings.trackerSettings['LINKING_MAX_DISTANCE'] = 8.0
83      settings.trackerSettings['GAP_CLOSING_MAX_DISTANCE'] = 8.0
84      settings.trackerSettings['MAX_FRAME_GAP'] = 1
85
86      # Configure track filters
87      settings.addTrackAnalyzer(TrackDurationAnalyzer())
```

```python
88      settings.addTrackAnalyzer(TrackSpotQualityFeatureAnalyzer())

89

90      filter1 = FeatureFilter('TRACK_DURATION', 20, True)
91      settings.addTrackFilter(filter1)

92

93      # Run TrackMate and store data into Model.
94      model = Model()
95      trackmate = TrackMate(model, settings)

96

97      ok = trackmate.checkInput()
98      if not ok:
99          sys.exit(str(trackmate.getErrorMessage()))

100

101     ok = trackmate.process()
102     if not ok:
103         sys.exit(str(trackmate.getErrorMessage()))

104

105     selectionModel = SelectionModel(model)
106     displayer =  HyperStackDisplayer(model, selectionModel, imp5)
107     displayer.render()
108     displayer.refresh()

109

110     IJ.log('TrackMate completed successfully.')
111     IJ.log('Found %d spots in %d tracks.' % (model.getSpots().getNSpots(True) ,
    model.getTrackModel().nTracks(True)))

112

113     # Print results in the console.
114     headerStr = '%10s %10s %10s %10s %10s %10s' % ('Spot_ID', 'Track_ID', 'Frame',
    'X', 'Y', 'Z')
115     rowStr = '%10d %10d %10d %10.1f %10.1f %10.1f'
116     for i in range( nChannels ):
117         headerStr += (' %10s' % ( 'C' + str(i+1) ) )
118         rowStr += ( ' %10.1f' )

119

120     #open a file to save results
121     myfile =
    open('/home/rickettsia/Desktop/data/Clamydial_Image_Analysis/
    EMS_BMECBMELVA_20X_01122019/data/'+orgtitle.split('.')[0]+'.csv', 'wb')
122     wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
123     wr.writerow(['Spot_ID', 'Track_ID', 'Frame', 'X', 'Y', 'Z', 'Channel_1',
    'Channel_2'])

124

125     IJ.log('\n')
126     IJ.log(headerStr)
127     tm = model.getTrackModel()
128     trackIDs = tm.trackIDs(True)
129     for trackID in trackIDs:
130         spots = tm.trackSpots(trackID)
```

```
131
132            # Let's sort them by frame.
133            ls = ArrayList(spots)
134
135            for spot in ls:
136                values = [spot.ID(), trackID, spot.getFeature('FRAME'), \
137                    spot.getFeature('POSITION_X'), spot.getFeature('POSITION_Y'),
    … spot.getFeature('POSITION_Z')]
138                for i in range(nChannels):
139                    values.append(spot.getFeature('MEAN_INTENSITY%02d' % (i+1)))
140
141                IJ.log(rowStr % tuple(values))
142                l1 = (values[0], values[1], values[2], values[3], values[4], values[5],
    … values[7], values[8])
143                wr.writerow(l1)
144
145        myfile.close()
146        IJ.selectWindow("Merged")
147        IJ.run("Close")
148
149
150 od = OpenDialog("Time Laps Images", "")
151 firstDir = od.getDirectory()
152 fileList = os.listdir(firstDir)
153
154 if "DisplaySettings.json" in fileList:
155     fileList.remove("DisplaySettings.json")
156 if ".DS_Store" in fileList:
157     fileList.remove(".DS_Store")
158
159 totalCount = []
160 i = 1
161 for fileName in fileList:
162     currentFile = firstDir + fileName
163     print(firstDir)
164     IJ.run("Bio-Formats Importer", "open=" + currentFile + " autoscale
    … color_mode=Composite view=Hyperstack stack_order=XYCZT")
165     track()
166
167
```

```
 93      IJ.selectWindow('Merged')
 94      IJ.run("Close")
 95
 96
 97  od = OpenDialog("Time Laps Images", "")
 98  firstDir = od.getDirectory()
 99  fileList = os.listdir(firstDir)
100
101  if "DisplaySettings.json" in fileList:
102      fileList.remove("DisplaySettings.json")
103  if ".DS_Store" in fileList:
104      fileList.remove(".DS_Store")
105  fileList.sort()
106  totalCount = []
107  i = 1
108  for fileName in fileList:
109      IJ.run("Collect Garbage")
110      currentFile = firstDir + fileName
111      print(firstDir)
112      IJ.run("Bio-Formats Importer", "open=[" + currentFile + "] autoscale
  …  color_mode=Composite view=Hyperstack stack_order=XYCZT series_list="+str(i))
113      reduceZ()
114      i=i+1
115
```

**Supplemental Material 4.S3: Clone Check**

# clone_check_Markdown

February 19, 2020

```
[ ]: %matplotlib notebook
     import matplotlib.pyplot as plt
     import numpy as np; np.random.seed(22)
     import seaborn as sns; sns.set(color_codes=True)
     import pandas as pd
     import math
```

```
[ ]: # Imports all csv files in folder and concatonates the data sets from trackmate.

     import glob

     df = pd.DataFrame()
     for filename in glob.glob('data2/*.csv'):
         data_01 = pd.read_csv(filename, sep=',')
         filename = filename.split('/')[1]
         filename = filename.split('.')[0]
         well = filename.split('_')[6]
         data_01['Well'] = well.split('-')[0]
         data_01['FOV'] = filename.split('_')[7]
         data = data_01[['Frame','Track_ID', 'Well', 'FOV', 'X', 'Y', 'Channel_2',␣
      ↪'Channel_3']]
         df = df.append(data,  ignore_index=True)
     df['Sample_ID'] = df.Well + '-' + df.FOV
     df['Sample_ID'] = df.Sample_ID + '-Track-' + df.Track_ID.astype(str)
```

```
[ ]: # Subtract baseline using min value: Channel_1.

     def subtract_bl(in_df):
         traces = in_df
         traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
      ↪values='Channel_2')
         df_test2 = pd.DataFrame()
         for columns in traces_p:
             minvalue = traces_p[columns].min()
             base_sub = lambda x: x-minvalue
             df_test = in_df[in_df['Sample_ID']==columns]
             df_test['bc_channel_1'] = df_test['Channel_2']-minvalue
```

```
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df)
```

```
[ ]: #subtract baseline using min value: Channel_2.

def subtract_bl(in_df):
    traces = in_df
    traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
 ↪values='Channel_3')
    df_test2 = pd.DataFrame()
    for columns in traces_p:
        minvalue = traces_p[columns].min()
        base_sub = lambda x: x-minvalue
        df_test = in_df[in_df['Sample_ID']==columns]
        df_test['bc_channel_2'] = df_test['Channel_3']-minvalue
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df_bl)
```

```
[ ]: # Save baseline subtracted data to pickle file.

df_bl.to_pickle("baseLine_subtract2.pkl")
```

```
[ ]: # Import baseline subtracted pickle file into Pandas dataframe.

df_bl = pd.read_pickle("baseLine_subtract2.pkl")
df_f=df_bl
```

```
[ ]: # Filter out inclusion near the edges of the field of view.

df2 = df_f[~(df_f['X']<10)]
df2 = df2[~(df2['X']>670)]
df2 = df2[~(df2['Y']<10)]
df2 = df2[~(df2['Y']>670)]
#df2 = df_f
```

```
[ ]: # Calibrate Frame values from the image slices to time values of experiment.

totalFrames = 79
startTime = 12
interval = 0.5

frame_dict = {}
for i in range(totalFrames):
```

```
    if i == 0:
        frame = i
        frame_dict[frame] = startTime+1
    else:
        frame = i
        startTime += interval
        frame_dict[frame] = startTime+1
```

```
[ ]: # Calibrate Frame values from the image slices to time values of experiment.

     df2['Time'] = df2['Frame'].map(frame_dict)
```

```
[ ]: # Filter out traces that do not extend over two time points.

     df_f1 = df2['Sample_ID'][df2['Time']==20]
     df_f2 = df2['Sample_ID'][df2['Time']==30]
     df_f3 = df_f1[df_f1.isin(df_f2)]
     df3 = df2[df2['Sample_ID'].isin(df_f3)]
```

```
[ ]: # Assign isolates to wells before filtering to MOI < 1.

     WT           = df3[df3['Sample_ID'].str.contains("B1-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C1-").fillna(False)]
     MUT_A2_8     = df3[df3['Sample_ID'].str.contains("B2-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C2-").fillna(False)]
     MUT_A3_1     = df3[df3['Sample_ID'].str.contains("B3-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C3-").fillna(False)]
     MUT_A3_6     = df3[df3['Sample_ID'].str.contains("B4-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C4-").fillna(False)]
     MUT_A3_7     = df3[df3['Sample_ID'].str.contains("B5-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C5-").fillna(False)]
     MUT_B1_7     = df3[df3['Sample_ID'].str.contains("B6-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C6-").fillna(False)]
     MUT_B1_9     = df3[df3['Sample_ID'].str.contains("B7-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C7-").fillna(False)]
     MUT_B1_11    = df3[df3['Sample_ID'].str.contains("B8-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C8-").fillna(False)]
     MUT_B2_10    = df3[df3['Sample_ID'].str.contains("B9-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("C9-").fillna(False)]
     MUT_B2_11    = df3[df3['Sample_ID'].str.contains("A10-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("B10-").fillna(False)]
     MUT_B2_11dot = df3[df3['Sample_ID'].str.contains("A11-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("B11-").fillna(False)]
     MUT_B3_6     = df3[df3['Sample_ID'].str.contains("A12-").
      ↪fillna(False)|df3['Sample_ID'].str.contains("B12-").fillna(False)]
```

3

```python
# Assign isolates to wells, MOI < 1.

WT            = df3[df3['Sample_ID'].str.contains("B1-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C1-").fillna(False)]
MUT_A2_8      = df3[df3['Sample_ID'].str.contains("C2-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C2-").fillna(False)]
MUT_A3_1      = df3[df3['Sample_ID'].str.contains("C3-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C3-").fillna(False)]
MUT_A3_6      = df3[df3['Sample_ID'].str.contains("C4-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C4-").fillna(False)]
MUT_A3_7      = df3[df3['Sample_ID'].str.contains("C5-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C5-").fillna(False)]
MUT_B1_7      = df3[df3['Sample_ID'].str.contains("C6-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C6-").fillna(False)]
MUT_B1_9      = df3[df3['Sample_ID'].str.contains("C7-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("C7-").fillna(False)]
MUT_B1_11     = df3[df3['Sample_ID'].str.contains("B8-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("B8-").fillna(False)]
MUT_B2_10     = df3[df3['Sample_ID'].str.contains("B9-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("B9-").fillna(False)]
MUT_B2_11     = df3[df3['Sample_ID'].str.contains("B10-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("B10-").fillna(False)]
MUT_B2_11dot  = df3[df3['Sample_ID'].str.contains("B11-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("B11-").fillna(False)]
MUT_B3_6      = df3[df3['Sample_ID'].str.contains("A12-").
 ↪fillna(False)]#[df3['Sample_ID'].str.contains("B12-").fillna(False)]
```

```python
# Filter for inclusions that exhibit sufficient growth.
# Set boolean for sample ID where Max value after time of min value is > 10␣
 ↪frames later.

def filterII(in_df, threshold):
    traces_p =        in_df.pivot_table(index='Frame', columns='Sample_ID',␣
 ↪values='bc_channel_2')
    df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
    for cn in traces_p.columns:
        index_min = traces_p[cn].idxmin(axis=1, skipna=True)
        min_value = traces_p[cn].min()
        traces_p_min = traces_p[traces_p.index>=index_min]
        max_value = traces_p_min[cn].max()
        late_value = traces_p[cn].iloc[54]
        ratio = late_value/max_value

        if max_value > threshold*(abs(min_value)+1) and ratio > 0.01:
            df_pass = df_pass.append({'Sample_ID':cn, 'pass': True},␣
 ↪ignore_index=True)
```

```
    new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
    return new_df

WT_f            =   filterII(WT           ,10000)
MUT_A2_8_f      =   filterII(MUT_A2_8     ,10000)
MUT_A3_1_f      =   filterII(MUT_A3_1     ,10000)
MUT_A3_6_f      =   filterII(MUT_A3_6     ,10000)
MUT_A3_7_f      =   filterII(MUT_A3_7     ,10000)
MUT_B1_7_f      =   filterII(MUT_B1_7     ,10000)
MUT_B1_9_f      =   filterII(MUT_B1_9     ,10000)
MUT_B1_11_f     =   filterII(MUT_B1_11    ,10000)
MUT_B2_10_f     =   filterII(MUT_B2_10    ,10000)
MUT_B2_11_f     =   filterII(MUT_B2_11    ,10000)
MUT_B2_11dot_f  =   filterII(MUT_B2_11dot ,10000)
MUT_B3_6_f      =   filterII(MUT_B3_6     ,10000)
```

```
[ ]: # Pivot dataframe use Time as index and Sample_ID as columns.
     # Calculate mean and standard deviation of each isolate.

     def pivot(in_df, channel):
         in_df_p = in_df.pivot_table(index='Time', columns='Sample_ID',␣
     ↪values=channel)
         in_df_p['mean'], in_df_p['std'] = in_df_p.mean(axis=1), in_df_p.std(axis=1)
         return in_df_p
```

```
[ ]: # Pivot dataframe use Time as index and Sample_ID as columns.

     EuoWT_p             = pivot(WT_f           ,  'bc_channel_1')
     EuoMUT_A2_8_p       = pivot(MUT_A2_8_f     ,  'bc_channel_1')
     EuoMUT_A3_1_p       = pivot(MUT_A3_1_f     ,  'bc_channel_1')
     EuoMUT_A3_6_p       = pivot(MUT_A3_6_f     ,  'bc_channel_1')
     EuoMUT_A3_7_p       = pivot(MUT_A3_7_f     ,  'bc_channel_1')
     EuoMUT_B1_7_p       = pivot(MUT_B1_7_f     ,  'bc_channel_1')
     EuoMUT_B1_9_p       = pivot(MUT_B1_9_f     ,  'bc_channel_1')
     EuoMUT_B1_11_p      = pivot(MUT_B1_11_f    ,  'bc_channel_1')
     EuoMUT_B2_10_p      = pivot(MUT_B2_10_f    ,  'bc_channel_1')
     EuoMUT_B2_11_p      = pivot(MUT_B2_11_f    ,  'bc_channel_1')
     EuoMUT_B2_11dot_p   = pivot(MUT_B2_11dot_f,  'bc_channel_1')
     EuoMUT_B3_6_p       = pivot(MUT_B3_6_f     ,  'bc_channel_1')


     HctBWT_p            = pivot(WT_f           ,  'bc_channel_2')
     HctBMUT_A2_8_p      = pivot(MUT_A2_8_f     ,  'bc_channel_2')
     HctBMUT_A3_1_p      = pivot(MUT_A3_1_f     ,  'bc_channel_2')
     HctBMUT_A3_6_p      = pivot(MUT_A3_6_f     ,  'bc_channel_2')
     HctBMUT_A3_7_p      = pivot(MUT_A3_7_f     ,  'bc_channel_2')
```

5

```
HctBMUT_B1_7_p        = pivot(MUT_B1_7_f    ,   'bc_channel_2')
HctBMUT_B1_9_p        = pivot(MUT_B1_9_f    ,   'bc_channel_2')
HctBMUT_B1_11_p       = pivot(MUT_B1_11_f   ,   'bc_channel_2')
HctBMUT_B2_10_p       = pivot(MUT_B2_10_f   ,   'bc_channel_2')
HctBMUT_B2_11_p       = pivot(MUT_B2_11_f   ,   'bc_channel_2')
HctBMUT_B2_11dot_p    = pivot(MUT_B2_11dot_f,   'bc_channel_2')
HctBMUT_B3_6_p        = pivot(MUT_B3_6_f    ,   'bc_channel_2')
```

[ ]: 
```python
#Check filtering threshold.

EuoMUT_B1_9_p.plot(legend=False)
EuoWT_p.plot(legend=False)
```

[ ]: 
```python
# Select color palette.

from matplotlib.ticker import MultipleLocator
from matplotlib.ticker import AutoMinorLocator
from matplotlib.ticker import LogLocator
c = sns.color_palette('Set1',16).as_hex()
c[1]
```

[ ]: 
```python
# Graph each isolate against wt strain using mean and SEM.

with plt.style.context('seaborn-white'):
    fig, ((ax2,ax3),(ax4,ax5),(ax6,ax7),(ax8,ax9),(ax10,ax11),(ax12,ax13)) =␣
 ↪plt.subplots(ncols=2, nrows=6)

    def plot_sample_2(sample, color, name, style, mstyle, fcolor, i):
        ax2.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
 ↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
 ↪markeredgecolor=color, markeredgewidth=1)
        ax2.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
 ↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
 ↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_2(EuoWT_p , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_2(HctBWT_p, c[1], 'HctB WT' , '--', '', 'None', 1)

    def plot_sample_3(sample, color, name, style, mstyle, fcolor, i):
        ax3.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
 ↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
 ↪markeredgecolor=color, markeredgewidth=1)
        ax3.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
 ↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
 ↪sqrt(len(sample.columns)), color=color, alpha=0.2)
```

```
    plot_sample_3(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_3(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_3(EuoMUT_A3_1_p , c[2], 'Euo MUT A3_1' , '-', '', 'None', 1)
    plot_sample_3(HctBMUT_A3_1_p, c[3], 'HctB MUT A3_1' , '-', '', 'None', 1)


def plot_sample_4(sample, color, name, style, mstyle, fcolor, i):
    ax4.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
    ax4.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_4(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_4(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_4(EuoMUT_A3_7_p , c[2], 'EuoMUT A3_7' , '-', '', 'None', 1)
    plot_sample_4(HctBMUT_A3_7_p, c[3], 'HctBMUT A3_7' , '-', '', 'None', 1)

def plot_sample_5(sample, color, name, style, mstyle, fcolor, i):
    ax5.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
    ax5.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_5(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_5(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_5(EuoMUT_A3_6_p , c[2], 'Euo MUT_A3_6' , '-', '', 'None', 1)
    plot_sample_5(HctBMUT_A3_6_p, c[3], 'HctB MUT_A3_6' , '-', '', 'None', 1)

def plot_sample_6(sample, color, name, style, mstyle, fcolor, i):
    ax6.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
    ax6.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_6(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_6(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_6(EuoMUT_B1_7_p , c[2], 'Euo MUT B1_7' , '-', '', 'None', 1)
    plot_sample_6(HctBMUT_B1_7_p, c[3], 'HctB MUT B1_7' , '-', '', 'None', 1)

def plot_sample_7(sample, color, name, style, mstyle, fcolor, i):
```

```python
        ax7.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
        ax7.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_7(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_7(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_7(EuoMUT_B1_9_p , c[2], 'Euo MUT B1_9' , '-', '', 'None', 1)
    plot_sample_7(HctBMUT_B1_9_p, c[3], 'HctB MUT B1_9' , '-', '', 'None', 1)

    def plot_sample_8(sample, color, name, style, mstyle, fcolor, i):
        ax8.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
        ax8.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_8(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_8(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_8(EuoMUT_A2_8_p , c[2], 'EuoMUT A2_5' , '-', '', 'None', 1)
    plot_sample_8(HctBMUT_A2_8_p, c[3], 'HctBMUT A2_5' , '-', '', 'None', 1)

    def plot_sample_9(sample, color, name, style, mstyle, fcolor, i):
        ax9.plot(sample.index, sample['mean']*i, color, label=name, linestyle =␣
↪style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
        ax9.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_9(EuoWT_p        , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_9(HctBWT_p       , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_9(EuoMUT_B1_11_p , c[2], 'Euo MUT B1_11' , '-', '', 'None', 1)
    plot_sample_9(HctBMUT_B1_11_p, c[3], 'HctB MUT B1_11' , '-', '', 'None', 1)

    def plot_sample_10(sample, color, name, style, mstyle, fcolor, i):
        ax10.plot(sample.index, sample['mean']*i, color, label=name, linestyle␣
↪= style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
        ax10.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)
```

```
    plot_sample_10(EuoWT_p         , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_10(HctBWT_p        , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_10(EuoMUT_B2_10_p , c[2], 'Euo MUT B2_10' , '-', '', 'None', 1)
    plot_sample_10(HctBMUT_B2_10_p, c[3], 'HctB MUT B2_10' , '-', '', 'None', 1)

  def plot_sample_11(sample, color, name, style, mstyle, fcolor, i):
      ax11.plot(sample.index, sample['mean']*i, color, label=name, linestyle␣
↪= style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
      ax11.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_11(EuoWT_p         , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_11(HctBWT_p        , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_11(EuoMUT_B2_11_p , c[2], 'Euo MUT B2_11' , '-', '', 'None', 1)
    plot_sample_11(HctBMUT_B2_11_p, c[3], 'HctB MUT B2_11' , '-', '', 'None', 1)

  def plot_sample_12(sample, color, name, style, mstyle, fcolor, i):
      ax12.plot(sample.index, sample['mean']*i, color, label=name, linestyle␣
↪= style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
      ax12.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_12(EuoWT_p          , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_12(HctBWT_p         , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_12(EuoMUT_B2_11dot_p , c[2], 'EuoMUT B2_11dot' , '-', '',␣
↪'None', 1)
    plot_sample_12(HctBMUT_B2_11dot_p, c[3], 'HctBMUT B2_11dot' , '-', '',␣
↪'None', 1)

  def plot_sample_13(sample, color, name, style, mstyle, fcolor, i):
      ax13.plot(sample.index, sample['mean']*i, color, label=name, linestyle␣
↪= style, marker=mstyle, alpha=0.8, ms = 5, markerfacecolor=fcolor,␣
↪markeredgecolor=color, markeredgewidth=1)
      ax13.fill_between(sample.index, sample['mean']*i-(sample['std']*i)/math.
↪sqrt(len(sample.columns)), sample['mean']*i+(sample['std']*i)/math.
↪sqrt(len(sample.columns)), color=color, alpha=0.2)

    plot_sample_13(EuoWT_p         , c[0], 'Euo WT' , '--', '', 'None', 1)
    plot_sample_13(HctBWT_p        , c[1], 'HctB WT' , '--', '', 'None', 1)
    plot_sample_13(EuoMUT_B3_6_p , c[2], 'Euo MUT B3_6' , '-', '', 'None', 1)
    plot_sample_13(HctBMUT_B3_6_p, c[3], 'HctB MUT B3_6' , '-', '', 'None', 1)
```

```python
with plt.style.context('classic'):
    ax2.legend(loc='upper left', fontsize=4)
    ax3.legend(loc='upper left', fontsize=4)
    ax4.legend(loc='upper left', fontsize=4)
    ax5.legend(loc='upper left', fontsize=4)
    ax6.legend(loc='upper left', fontsize=4)
    ax7.legend(loc='upper left', fontsize=4)
    ax8.legend(loc='upper left', fontsize=4)
    ax9.legend(loc='upper left', fontsize=4)
    ax10.legend(loc='upper left', fontsize=4)
    ax11.legend(loc='upper left', fontsize=4)
    ax12.legend(loc='upper left', fontsize=4)
    ax13.legend(loc='upper left', fontsize=4)
    ax2.set_ylabel('Relative Fluorescence')
    ax2.set_xlabel('HPI')
    ax3.set_xlabel('HPI')
    ax4.set_xlabel('HPI')
    ax5.set_xlabel('HPI')
    ax6.set_xlabel('HPI')
    ax7.set_xlabel('HPI')
    ax8.set_xlabel('HPI')
    ax9.set_xlabel('HPI')
    ax10.set_xlabel('HPI')
    ax11.set_xlabel('HPI')
    ax12.set_xlabel('HPI')
    ax13.set_xlabel('HPI')

    #ax2.set_xlim([12, 48])
    #ax3.set_xlim([12, 48])
    #ax4.set_xlim([12, 48])


    fig.set_size_inches(6, 24)
    #plt.savefig('MutantVerification.pdf')
```

```python
[ ]: # Calculate max, min, halfmax and time to halfmax.

def halfmax(in_pd):
    in_pd_c1_p = in_pd.pivot_table(index='Sample_ID', columns='Time',
 ↪values='bc_channel_1')
    in_pd_c1_p['max_c1'] = in_pd_c1_p.max(axis=1)
    in_pd_c2_p = in_pd.pivot_table(index='Sample_ID', columns='Time',
 ↪values='bc_channel_2')
    in_pd_c2_p['max_c2'] = in_pd_c2_p.max(axis=1)

    in_pd_c1_p['min_c1'] = 1
    in_pd_c2_p['min_c2'] = 1
```

```python
    in_pd_c1_p['half_max_c1'] = in_pd_c1_p['max_c1']/2
    in_pd_c2_p['half_max_c2'] = in_pd_c2_p['max_c2']/2

    in_pd_f = in_pd_c1_p[['min_c1','max_c1', 'half_max_c1']]
    in_pd_f['min_c2'] = in_pd_c2_p['min_c2']
    in_pd_f['max_c2'] = in_pd_c2_p['max_c2']
    in_pd_f['half_max_c2'] = in_pd_c2_p['half_max_c2']
    in_pd_f.index.name = None
    in_pd_f['Sample_ID'] = in_pd_f.index


    traces = in_pd

    traces_c1_p =        traces.pivot_table(index='Time', columns='Sample_ID',␣
↪values='bc_channel_1')
    traces_c1_p_fill =  traces_c1_p.fillna(method='bfill') #back fill to frame 0
    traces_c1_p_fill.reset_index(inplace = True)
    df_c1_f = traces_c1_p_fill
    df_c1_f.set_index(df_c1_f['Time'], inplace=True)

    traces_c2_p =        traces.pivot_table(index='Time', columns='Sample_ID',␣
↪values='bc_channel_2')
    traces_c2_p_fill =  traces_c2_p.fillna(method='bfill') #back fill to frame 0
    traces_c2_p_fill.reset_index(inplace = True)
    df_c2_f = traces_c2_p_fill
    df_c2_f.set_index(df_c2_f['Time'], inplace=True)

    time_halfmax_c1 = pd.DataFrame(columns=['Sample_ID', 'time_to_halfmax_c1'])
    time_halfmax_c2 = pd.DataFrame(columns=['Sample_ID', 'time_to_halfmax_c2'])
    for index, row in in_pd_f.iterrows():
        sampleID = row['Sample_ID']

        time_c1 = (df_c1_f[sampleID]-row['half_max_c1']).abs().sort_values().
↪index[0]
        time_c2 = (df_c2_f[sampleID]-row['half_max_c2']).abs().sort_values().
↪index[0]

        time_halfmax_c1 = time_halfmax_c1.append({'Sample_ID':sampleID,␣
↪'time_to_halfmax_c1': time_c1}, ignore_index=True)
        time_halfmax_c2 = time_halfmax_c2.append({'Sample_ID':sampleID,␣
↪'time_to_halfmax_c2': time_c2}, ignore_index=True)

    data_summary = pd.merge(time_halfmax_c1, time_halfmax_c2, on='Sample_ID')
    # Add x y coordinates at last time imaged.
    df_time = df3[df3['Time']==35]
    df_time = df_time[['Sample_ID', 'X', 'Y']]
    # Data_summary only for Sample IDs in both
```

```
        result = pd.merge(data_summary, df_time, how='inner', on=['Sample_ID'])
        return result


WT_hm              = halfmax(WT_f          )

MUT_A2_8_hm        = halfmax(MUT_A2_8_f     )
MUT_A3_1_hm        = halfmax(MUT_A3_1_f     )
MUT_A3_6_hm        = halfmax(MUT_A3_6_f     )
MUT_A3_7_hm        = halfmax(MUT_A3_7_f     )
MUT_B1_7_hm        = halfmax(MUT_B1_7_f     )
MUT_B1_9_hm        = halfmax(MUT_B1_9_f     )
MUT_B1_11_hm       = halfmax(MUT_B1_11_f    )
MUT_B2_10_hm       = halfmax(MUT_B2_10_f    )
MUT_B2_11_hm       = halfmax(MUT_B2_11_f    )
MUT_B2_11dot_hm    = halfmax(MUT_B2_11dot_f )
MUT_B3_6_hm        = halfmax(MUT_B3_6_f     )
```

```
[ ]:  # Visualize idividual traces in interactive graph.

      from bokeh.models import ColumnDataSource
      from bokeh.plotting import figure
      from bokeh.io import push_notebook, show, output_notebook, output_file
      from bokeh.layouts import row
      from bokeh.models import ColumnDataSource, Range1d, LabelSet, Label
      from bokeh.models import HoverTool
      from collections import OrderedDict
      from bokeh.layouts import layout
      from bokeh.models import Toggle, BoxAnnotation, CustomJS

      output_notebook()

      c = sns.color_palette('Set1',1000).as_hex()

      def interactive_graph(in_df):
          in_df_p = pivot(in_df, 'bc_channel_2')
          df_new = in_df_p[in_df_p.columns[0:-2]]
          result = halfmax(in_df)

          tools_to_show = 'box_zoom,save,hover,reset'
          p = figure(plot_height = 500, plot_width = 500,
                     toolbar_location='above', tools=tools_to_show,
                     # "easy" tooltips in Bokeh 0.13.0 or newer
                     tooltips=[("Location","$name")])
          j=0
          for i, column in enumerate(df_new):
              trackx = result[result['Sample_ID'] == df_new.iloc[:,i].name].X.values
```

```
        tracky = result[result['Sample_ID'] == df_new.iloc[:,i].name].Y.values
        x = str(int(trackx))
        y = str(int(tracky))
        well = df_new.iloc[:,i].name.split('-')[0]
        field = df_new.iloc[:,i].name.split('-')[1]
        well = df_new.iloc[:,i].name
        name = well+', x='+x+', y='+y
        p.line(df_new.index.values, df_new.iloc[:,i].values, name = name,␣
↪line_color=c[j], line_width=2.5)
        j=j+1
    p.legend.location = "top_left"
    p.legend.click_policy="mute"
    show(p)
interactive_graph(MUT_B3_6_f )
```

```
[ ]: # Visualize time to half-maximal expression of each promoter for individual␣
     ↪inclusions.
     # Identify location of inclusion outliers.

     def spot_plot(title, in_df, in_df2):
         TOOLTIPS = [("ID", "@Sample_ID"), ('X', '@X'), ('Y', '@Y')]
         p = figure(title=title, tooltips=TOOLTIPS, y_range=(15, 60),␣
     ↪x_range=(15,60), plot_width=600, plot_height=300)
         source = ColumnDataSource(in_df)
         source2= ColumnDataSource(in_df2)
         p.circle(x='time_to_halfmax_c1', y='time_to_halfmax_c2', size=10,␣
     ↪color='green', alpha=0.25, source=source)
         p.circle(x='time_to_halfmax_c1', y='time_to_halfmax_c2', size=10,␣
     ↪color='blue', alpha=0.25, source=source2)
         show(p)

     spot_plot('EMS', MUT_B2_11_hm, WT_hm)
```

```
[ ]: # Visualize promoter expression kientics of each inclusion through time.

     WT_f['treatment']='none'
     MUT_B2_11_f['treatment']='EMS'
     df3 = pd.concat([WT_f, MUT_B2_11_f])
     df3.sort_values('Time', inplace=True)

     import plotly.express as px
     gapminder = px.data.gapminder()
     fig = px.scatter(df3, x="bc_channel_1", y="bc_channel_2",␣
      ↪animation_frame="Time",
                 hover_name="Sample_ID", animation_group="Sample_ID",␣
      ↪color='treatment',
```

13

```
          log_x=False, size_max=85, range_x=[0, 60000], range_y=[-500, 20000],␣
 ↪color_discrete_map = {"none": 'rgba(0, 0, 255, .3)', "EMS": 'rgba(0, 255, 0,␣
 ↪.3)'})
fig.update_traces(marker=dict(size=6))
fig.show()
```

[ ]:
```
# Visualize euo vs hctB expression at 32 HPI for individual inclusions.
# Identify location of inclusion outliers.

MT32 = MUT_B3_6_f[MUT_B3_6_f['Time']==32] # Get values at specified HPI for all␣
 ↪sample IDs.
WT32 = WT_f[WT_f['Time']==32] # Get values at specified HPI for all sample IDs.
def spot_plot(title, in_df, in_df2):
    TOOLTIPS = [("ID", "@Sample_ID"), ('X', '@X'), ('Y', '@Y')]
    p = figure(title=title, tooltips=TOOLTIPS, x_range=(0, 60000),␣
 ↪y_range=(-1000, 12000), plot_width=600, plot_height=300)
    source = ColumnDataSource(in_df)
    source2= ColumnDataSource(in_df2)
    p.circle(x='bc_channel_1', y='bc_channel_2', size=10, color='green',␣
 ↪alpha=0.25, source=source)
    p.circle(x='bc_channel_1', y='bc_channel_2', size=10, color='blue', alpha=0.
 ↪25, source=source2)
    show(p)

spot_plot('EUO vs HctB @32 HPI', MT32, WT32)
```

**Supplemental Material 4.S4: EMS Screen**

# EMS_Screen-Markdown

February 19, 2020

```
[ ]: %matplotlib notebook
     import matplotlib.pyplot as plt
     import numpy as np; np.random.seed(22)
     import seaborn as sns; sns.set(color_codes=True)
     import pandas as pd
     import math
```

```
[ ]: # Imports all csv files in folder and concatonates the data sets from trackmate.

     import glob

     df = pd.DataFrame()
     for filename in glob.glob('data/*.csv'):
         data_01 = pd.read_csv(filename, sep=',')
         filename = filename.split('/')[1]
         filename = filename.split('.')[0]
         well = filename.split('_')[6]
         data_01['Well'] = well.split('-')[0]
         data_01['FOV'] = filename.split('_')[7]
         data = data_01[['Frame','Track_ID', 'Well', 'FOV', 'X', 'Y', 'Channel_2',␣
      ↪'Channel_3']]
         df = df.append(data,  ignore_index=True)
     df['Sample_ID'] = df.Well + '-' + df.FOV
     df['Sample_ID'] = df.Sample_ID + '-Track-' + df.Track_ID.astype(str)
```

```
[ ]: # Subtract baseline using min value: Channel_1.

     def subtract_bl(in_df):
         traces = in_df
         traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
      ↪values='Channel_2')
         df_test2 = pd.DataFrame()
         for columns in traces_p:
             minvalue = traces_p[columns].min()
             base_sub = lambda x: x-minvalue
             df_test = in_df[in_df['Sample_ID']==columns]
             df_test['bc_channel_1'] = df_test['Channel_2']-minvalue
```

1

```
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df)
```

```
[ ]: #subtract baseline using min value: Channel_2.

def subtract_bl(in_df):
    traces = in_df
    traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
    ↪values='Channel_3')
    df_test2 = pd.DataFrame()
    for columns in traces_p:
        minvalue = traces_p[columns].min()
        base_sub = lambda x: x-minvalue
        df_test = in_df[in_df['Sample_ID']==columns]
        df_test['bc_channel_2'] = df_test['Channel_3']-minvalue
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df_bl)
```

```
[ ]: # Save baseline subtracted data to pickle file.

df_bl.to_pickle("baseLine_subtract.pkl")
```

```
[ ]: # Import baseline subtracted pickle file into Pandas dataframe.

df_bl = pd.read_pickle("baseLine_subtract.pkl")
df_f=df_bl
```

```
[ ]: # Filter out inclusion near the edges of the field of view.

df2 = df_f[~(df_f['X']<10)]
df2 = df2[~(df2['X']>670)]
df2 = df2[~(df2['Y']<10)]
df2 = df2[~(df2['Y']>670)]
```

```
[ ]: # Calibrate Frame values from the image slices to time values of experiment.

totalFrames = 49
startTime = 12
interval = 0.5

frame_dict = {}
for i in range(totalFrames):
    if i == 0:
```

```
            frame = i
            frame_dict[frame] = startTime+1
        else:
            frame = i
            startTime += interval
            frame_dict[frame] = startTime+1
df2['Time'] = df2['Frame'].map(frame_dict)
```

```
[ ]: # Filter out traces that do not extend over two time points.

     df_f1 = df2['Sample_ID'][df2['Time']==15]
     df_f2 = df2['Sample_ID'][df2['Time']==35]
     df_f3 = df_f1[df_f1.isin(df_f2)]
     df3 = df2[df2['Sample_ID'].isin(df_f3)]
```

```
[ ]: # Assign treatments to wells, Wt = wildtype, Mt = mutant.

     Wt     = df3[df3['Well'].str.contains("A1").fillna(False)]
     Mt     = df3[df3['Well'].str.contains("A2").fillna(False)|df3['Well'].str.
      ↪contains("A3").fillna(False)|
                                                         df3['Well'].str.
      ↪contains("B1").fillna(False)|
                                                         df3['Well'].str.
      ↪contains("B2").fillna(False)|
                                                         df3['Well'].str.
      ↪contains("B3").fillna(False)]
```

```
[ ]: # Filter for inclusions that exhibit sufficient growth.
     # Set boolean for sample ID where Max value after time of min value is > 10␣
      ↪frames later.

     def filterII(in_df, threshold):
         traces_p =        in_df.pivot_table(index='Frame', columns='Sample_ID',␣
      ↪values='bc_channel_1')
         df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
         for cn in traces_p.columns:
             index_min = traces_p[cn].idxmin(axis=1, skipna=True)
             min_value = traces_p[cn].min()
             traces_p_min = traces_p[traces_p.index>=index_min]
             max_value = traces_p_min[cn].max()

             if max_value > threshold*(abs(min_value)+1):
                 df_pass = df_pass.append({'Sample_ID':cn, 'pass': True},␣
      ↪ignore_index=True)

         new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
```

```
        return new_df

Wt_f=  filterII(Wt,10000)
Mt_f=  filterII(Mt,10000)
```

```
[ ]: # Pivot dataframe use Time as index and Sample_ID as columns.

     def pivot(in_df, channel):
         in_df_p = in_df.pivot_table(index='Time', columns='Sample_ID',␣
      ↪values=channel)
         in_df_p['mean'], in_df_p['std'] = in_df_p.mean(axis=1), in_df_p.std(axis=1)
         return in_df_p
```

```
[ ]: # Pivot dataframe use Time as index and Sample_ID as columns.

     Wt_Euo_p     = pivot(Wt_f,   'bc_channel_1')
     Wt_HctB_p    = pivot(Wt_f,   'bc_channel_2')

     Mt_Euo_p     = pivot(Mt_f,   'bc_channel_1')
     Mt_HctB_p    = pivot(Mt_f,   'bc_channel_2')
```

```
[ ]: #Check filtering threshold.

     Wt_Euo_p.plot(legend=False)
```

```
[ ]: # Count number of inclusion tracks.

     Wt_inclusion_number = Wt_Euo_p.shape[1]-2
     Mt_inclusion_number = Mt_Euo_p.shape[1]-2
     print('Number of Inclusion Tracks =', Wt_inclusion_number + Mt_inclusion_number)
```

```
[ ]: # Calculate percent mortality.

     mt_tracks = Mt_Euo_p.shape[1]-2
     wt_tracks = Wt_Euo_p.shape[1]-2
     print('Percent Mortality =' ,100-(mt_tracks/5)/(wt_tracks*3)*100)
```

```
[ ]: # Calculate max, min, halfmax and time to halfmax.

     def halfmax(in_pd):
         in_pd_c1_p = in_pd.pivot_table(index='Sample_ID', columns='Time',␣
      ↪values='bc_channel_1')
         in_pd_c1_p['max_c1'] = in_pd_c1_p.max(axis=1)
         in_pd_c2_p = in_pd.pivot_table(index='Sample_ID', columns='Time',␣
      ↪values='bc_channel_2')
         in_pd_c2_p['max_c2'] = in_pd_c2_p.max(axis=1)
         in_pd_c1_p['min_c1'] = 1
```

4

```python
    in_pd_c2_p['min_c2'] = 1
    in_pd_c1_p['half_max_c1'] = in_pd_c1_p['max_c1']/2
    in_pd_c2_p['half_max_c2'] = in_pd_c2_p['max_c2']/2
    in_pd_f = in_pd_c1_p[['min_c1','max_c1', 'half_max_c1']]
    in_pd_f['min_c2'] = in_pd_c2_p['min_c2']
    in_pd_f['max_c2'] = in_pd_c2_p['max_c2']
    in_pd_f['half_max_c2'] = in_pd_c2_p['half_max_c2']
    in_pd_f.index.name = None
    in_pd_f['Sample_ID'] = in_pd_f.index
    traces = in_pd
    traces_c1_p =        traces.pivot_table(index='Time', columns='Sample_ID',␣
↪values='bc_channel_1')
    traces_c1_p_fill =  traces_c1_p.fillna(method='bfill') #back fill to frame 0
    traces_c1_p_fill.reset_index(inplace = True)
    df_c1_f = traces_c1_p_fill
    df_c1_f.set_index(df_c1_f['Time'], inplace=True)
    traces_c2_p =        traces.pivot_table(index='Time', columns='Sample_ID',␣
↪values='bc_channel_2')
    traces_c2_p_fill =  traces_c2_p.fillna(method='bfill') #back fill to frame 0
    traces_c2_p_fill.reset_index(inplace = True)
    df_c2_f = traces_c2_p_fill
    df_c2_f.set_index(df_c2_f['Time'], inplace=True)
    time_halfmax_c1 = pd.DataFrame(columns=['Sample_ID', 'time_to_halfmax_c1'])
    time_halfmax_c2 = pd.DataFrame(columns=['Sample_ID', 'time_to_halfmax_c2'])
    for index, row in in_pd_f.iterrows():
        sampleID = row['Sample_ID']
        time_c1 = (df_c1_f[sampleID]-row['half_max_c1']).abs().sort_values().
↪index[0]
        time_c2 = (df_c2_f[sampleID]-row['half_max_c2']).abs().sort_values().
↪index[0]

        time_halfmax_c1 = time_halfmax_c1.append({'Sample_ID':sampleID,␣
↪'time_to_halfmax_c1': time_c1}, ignore_index=True)
        time_halfmax_c2 = time_halfmax_c2.append({'Sample_ID':sampleID,␣
↪'time_to_halfmax_c2': time_c2}, ignore_index=True)


    data_summary = pd.merge(time_halfmax_c1, time_halfmax_c2, on='Sample_ID')
    # Add x y coordinates at last time imaged.
    df_time = df3[df3['Time']==35]
    df_time = df_time[['Sample_ID', 'X', 'Y']]
    # Data_summary   only for Sample IDs in both.
    result = pd.merge(data_summary, df_time, how='inner', on=['Sample_ID'])
    return result

Wt_hm = halfmax(Wt_f)
```

5

```
Mt_hm = halfmax(Mt_f)
```

```python
# Visualize idividual traces in interactive graph.

from bokeh.models import ColumnDataSource
from bokeh.plotting import figure
from bokeh.io import push_notebook, show, output_notebook, output_file
from bokeh.layouts import row
from bokeh.models import ColumnDataSource, Range1d, LabelSet, Label
from bokeh.models import HoverTool
from collections import OrderedDict
from bokeh.layouts import layout
from bokeh.models import Toggle, BoxAnnotation, CustomJS


output_notebook()

c = sns.color_palette('Set1',1000).as_hex()

def interactive_graph(in_df):
    in_df_p = pivot(in_df, 'bc_channel_1')
    df_new = in_df_p[in_df_p.columns[0:-2]]
    result = halfmax(in_df)
    #result = in_df

    tools_to_show = 'box_zoom,save,hover,reset'
    p = figure(plot_height = 500, plot_width = 500,
               toolbar_location='above', tools=tools_to_show,
               # "easy" tooltips in Bokeh 0.13.0 or newer
               tooltips=[("Location","$name")])
    j=0
    for i, column in enumerate(df_new):
        trackx = result[result['Sample_ID'] == df_new.iloc[:,i].name].X.values
        tracky = result[result['Sample_ID'] == df_new.iloc[:,i].name].Y.values
        x = str(int(trackx))
        y = str(int(tracky))
        well = df_new.iloc[:,i].name.split('-')[0]
        field = df_new.iloc[:,i].name.split('-')[1]
        well = df_new.iloc[:,i].name
        name = well+', x='+x+', y='+y
        p.line(df_new.index.values, df_new.iloc[:,i].values, name = name,␣
 ↪line_color=c[j], line_width=2.5)
        j=j+1
    p.legend.location = "top_left"
    p.legend.click_policy="mute"
    show(p)
interactive_graph(Mt_f)
```

```python
[ ]: # Visualize time to half-maximal expression of each promoter for individual␣
     ↪inclusions.
     # Identify location of inclusion outliers.

     def spot_plot(title, in_df, in_df2):
         TOOLTIPS = [("ID", "@Sample_ID"), ('X', '@X'), ('Y', '@Y')]
         p = figure(title=title, tooltips=TOOLTIPS, y_range=(29, 38), x_range=(15,␣
     ↪35), plot_width=600, plot_height=300)
         source = ColumnDataSource(in_df)
         source2= ColumnDataSource(in_df2)
         p.circle(x='time_to_halfmax_c1', y='time_to_halfmax_c2', size=10,␣
     ↪color='red', alpha=0.25, source=source)
         p.circle(x='time_to_halfmax_c1', y='time_to_halfmax_c2', size=10,␣
     ↪color='blue', alpha=0.25, source=source2)
         show(p)

     spot_plot('EMS', Mt_hm, Wt_hm)
```

```python
[ ]: # Visualize promoter expression kientics of each inclusion through time.

     Wt_f['treatment']='none'
     Mt_f['treatment']='EMS'
     df3 = pd.concat([Wt_f, Mt_f])
     df3.sort_values('Time', inplace=True)
     df3['euoprom']= df3["bc_channel_1"]
     df3['hctBprom']= df3["bc_channel_2"]

     import plotly.express as px

     gapminder = px.data.gapminder()
     fig = px.scatter(df3, x='euoprom', y='hctBprom', animation_frame="Time",
                 hover_name="Sample_ID", animation_group="Sample_ID",␣
      ↪color='treatment',
                 log_x=False, size_max=85, range_x=[-500, 55000], range_y=[-100,␣
     ↪9000], color_discrete_map = {"none": 'rgba(0, 0, 255, .3)', "EMS":␣
     ↪'rgba(255, 50, 50, .3)'})
     fig.update_traces(marker=dict(size=10))
     fig.show()
```

```python
[ ]: # Visualize euo vs hctB expression at 32 HPI for individual inclusions.
     # Identify location of inclusion outliers.

     Mt28 = Mt_f[Mt_f['Time']==28] # Get values at specified HPI for all sample IDs.
     Wt28 = Wt_f[Wt_f['Time']==28] # Get values at specified HPI for all sample IDs.
     def spot_plot(title, in_df, in_df2):
         TOOLTIPS = [("ID", "@Sample_ID"), ('X', '@X'), ('Y', '@Y')]
```

```
    p = figure(title=title, tooltips=TOOLTIPS, x_range=(0, 45000),␣
↪y_range=(-100, 1000), plot_width=600, plot_height=300)
    source = ColumnDataSource(in_df)
    source2= ColumnDataSource(in_df2)
    p.circle(x='bc_channel_1', y='bc_channel_2', size=10, color='red', alpha=0.
↪25, source=source)
    p.circle(x='bc_channel_1', y='bc_channel_2', size=10, color='blue', alpha=0.
↪25, source=source2)
    show(p)

spot_plot('EUO vs HctB @28 HPI', Mt28, Wt28)
```

**Table 4.1**

| Name of Material/ Equipment | Company | Catalog/Part Number | Comments/ Description |
|---|---|---|---|
| 24-well polystyrene plates | Corning | 3524 | Cell culture growth for reinfection of isolates |
| 6-well glass bottom plates | Cellvis | P06-1.5H-N | Cell culture growth for imaging |
| 96-well glass bottom plates | Nunc | 165305 | Cell culture growth for imaging |
| Bold line CO2 Unit | OKO Labs | CO2 UNIT BL | Stage incubator CO2 control |
| Bold line T Unit | OKO Labs | H301-T-UNIT-BL-PLUS | Stage incubator temperature control |
| Borosilicate glass capillary tubes | Sutter Instrument | B1005010 | Capillary tubes |
| BrightLine bandpass emissions filter (514/30nm) | Semrock | FF01-514/30-25 | Fluoescent filter cube |
| BrightLine bandpass emissions filter (641/75nm) | Semrock | FF02-641/75-25 | Fluorescent filter cube |
| CellTram Vario | Eppendorf | 5196000030 | Microinjector |
| *Chlamydia trachomatis* serovar L2 | ATCC | VR-577 | *Chlamydia trachomatis* |

| CIP-1 media | In house | NA | Axenic media. IPB supplemented with 1% FBS, 25 µM amino acids, 0.5mM G6P, 1.0 mM ATP, 0.5 mM DTT, and 50 µM GTP, UTP, and CTP. (Omsland, A. 2012) made in-house. |
|---|---|---|---|
| Cos-7 cells (ATCC) | ATCC | CRL-1651 | African green monkey kidney cell (host cells) |
| Cycloheximide | M P Biomedicals | 194527 | Host cell growth inhibitor |
| Ethyl methanesulfonate, 99% | Acros Organics | AC205260100 | Mutagen |
| Fetal Plex | Gemini Bio- Products | 100-602 | Supplement for base growth media |
| Fiji/ImageJ | https://imagej.net/Fiji | NA | Open source Image analysis software. https://imagej.net/Fiji |
| Galaxy 170 S CO2 incubator | Eppendorf | CO1700100X | Cell culture incubation |
| gblocks (Fluorescent FP variants: Clover and mKate2) | Integrated DNA Technologies | NA | gblock ORFs of Ctr optimized FP varients for cloning into p2TK2SW2 |
| Gentamycin 10mg/ml | Gibco | 15710-064 | Antibiotic for growth media |

| HBSS (Hank's Balanced Salt Solution) | Corning | 21-020-CM | Host cells rinse |
|---|---|---|---|
| Heparin sodium | Amersham Life Science | 16920 | inhibits and reverses the early electrostatic interactions between the host cell and EBs |
| HEPES 1M | GE Life Sciences | SH30237.01 | pH buffer for growth media |
| InjectMan | Eppendorf | 5179 000.018 | Micromanipulator |
| Jupyter Notebook | https://jupyter.org/ | NA | Visualization of inclusion traces. https://jupyter.org/ |
| Lambda 10-3 | Sutter Instrument | LB10-3 | Filter wheel controller |
| Oko Touch | OKO Labs | Oko Touch | Interface to control the Bold line T and CO2 Unit |
| Prior XY stage | Prior | H107 | Motorized XY microscope stage |
| PrismR Centrifuge | Labnet | C2500-R | Temperature controlled microcentrifuge |
| Problot Hybridization oven | Labnet | H1200A | Rocking Incubator for infection with *Chlamydia* |
| Proscan II | Prior | H30V4 | XYZ microscope stage controller |
| Purifier Class 2 Biosafety Cabinet | Labconco | 362804 | Cell culture work |

| RPMI-1640 (no phenol red) | Gibco | 11835-030 | Base growth media for imaging |
|---|---|---|---|
| RPMI-1640 (phenol red) | GE Life Sciences | SH30027.01 | Base growth media |
| scopeLED excitation LEDs (470nm,595nm) | scopeLED | F140 | Excitation light |
| Sonic Dismembrator Model 500 | Fisher Scientific | 15-338-550 | Sonicator, resuspending chlamydial pellet |
| Stage incubator | OKO Labs | H301-K-FRAME | Cluster well plate incubation chamber |
| sucrose-phosphate-glutamate buffer 1X (SPG) | In house | NA | Chlamydial storage buffer. (10 mM sodium phosphate [8 mM $K_2HPO_4$, 2 mM $KH_2PO_4$], 220 mM sucrose, 0.50 mM L-glutamic acid; pH 7.4) |
| T-75 Flasks | Thermo Scientific | 156499 | Cell culture growth |
| TE 300 inverted microscope | Nikon | 16724 | microscope |
| THOR LED | Thor Labs | LEDD1B | White light |
| Trypsin | Corning | 25-052-CI | Dislodges host cells from flask for seeding into plates |
| Zyla sCMOS | Andor | ZYLA-5.5-USB3 | imaging camera |

| | | | |
|---|---|---|---|
| μ Manager 2.0gamma | https://github.com/micro-manager/micro-manager | NA | Open source automated microscope control software package |

CHAPTER FIVE: UNPUBLISHED RESEARCH

This chapter contains more recent unpublished *in vivo* experiments. The data from this chapter has expanded our knowledge of cell-form development in regards to the mechanism of $RB_R$-to-$RB_E$ maturation as well as the rate chlamydial replication and EB production.

**RB maturation is independent of RB numbers and cell division.**

The variation in *euo*prom fluorescent intensity and RB numbers within individual inclusions from automated live-cell and fixed confocal experiments described in **Chapters 2 and 3** (**Mov. 2.S2, Fig. 3.2, 3.8**) suggested that $RB_R$-to-$RB_E$ maturation varies from inclusion to inclusion. Therefore, to better characterize $RB_E$ dynamics, we quantified the number of RBs and IBs within individual inclusions using the *Ctr-L2-euo*prom-mNG(LVA)_*hctA*prom-mKate2 (RB/IB) promoter-reporter strain. Host cells were infected at an MOI of 0.1 and samples were treated with penicillin in a time series from 10-17 hpi (every hour). Penicillin treatment remained on samples for 4 h to prevent further chlamydial replication and allow for maximum *hctA*prom-mKate2 expression; after which samples were fixed and stained with DAPI to label the host and chlamydial DNA. Chlamydial cells positive for *euo*prom-mNG(LVA) and DAPI were counted as RBs, whereas *hctA*prom-mKate2+/DAPI+ cells were counted as IBs (**Fig. 5.1A**). RB and IB numbers were quantified manually on a per-inclusion basis using the multi-point tool in Fiji.

Intra-inclusion RB numbers increased from 10-17 hpi (**Fig. 5.1B**). Inclusions containing at least one IB, and therefore one $RB_E$, were present at all time points. Whereas, inclusions absent of IBs (no $RB_E$s) were observed up to 16 hpi. From 13-17 hpi, per-inclusion IB numbers demonstrated an ever increasing range with inclusions from the 17 hpi time point presenting with the largest variability at 1 to >50 IBs (**Fig. 5.1C**). IB presence was seen in inclusions that contained as little as 2 RBs (**Fig. 5.1D**). Although the increase in IB numbers correlated with the increase in RB amplification, there was again large variation in the number of IBs per RB replication number (**Fig. 5.1E**). The range of RB numbers varied widely across all time points,

similar to the variation seen in *euo*prom kinetics and RB numbers from **Chapters 2 and 3.**

The presence of IBs across all time points, regardless of RB number, suggests that $RB_E$ maturation is independent of both total number of RBs as well as cell division. Although these results rule out RB and replication number as factors that regulate $RB_E$ maturation, more research is needed to determine actual $RB_R$-to-$RB_E$ regulatory mechanism.

**EB production is dependent on total RB numbers, and chlamydial replication is constant.**

Data from **Chapters 2 and 3** showed that the rate of EB production (*hctB*prom-mKate2) varies greatly between inclusions (**Mov. 2.S2, Fig. 3.5**)**.** This phenomenon appears to occur on a per-inclusion basis with the slope of *hctB*prom-mKate2 expression corresponding directly with the plateau level of *euo*prom-Clover (**Mov. 2.S2**). In **Chapter 3**, we showed that the plateau in *euo*prom expression correlated with the number of RBs within an individual inclusion (**Fig. 3.2**). To determine if EB production was dependent on the total number of RBs, *Ctr*-L2-*euo*prom-Clover_*hctB*prom-mKate2 was used to infect host cells at an MOI of 0.1 and the fluorescent intensity of *euo*prom-Clover and *hctB*prom-mKate2 were monitored from 10-80 hpi with automated live-cell microscopy (**Fig. 5.2AB**). After fluorescent intensity quantification of both reporters, the levels of *euo*prom-Clover during the plateau period (50-60 hpi) from individual inclusion traces were normalized with custom python scripts (**Fig. 5.2C**) (**Supplemental Material 5.S1**). The individual inclusion normalization factor of each inclusion was then used to adjust the levels *hctB*prom-mKate2 expression from the respective inclusion. Surprisingly, this normalization technique collapsed the inter-inclusion variation in *hctB*prom-mKate2 expression, producing near equal EB production slopes across inclusions (**Fig. 5.2D**). In agreement with **Fig. 3.5** and **3.8**, this data suggested that the total number of RBs is stable during the plateau period, yet differs between inclusions. EB production rates also appeared to be dependent on total RB numbers, however, normalizing *hctB*prom-mKate2 expression revealed that *Chlamydia* within isolated

inclusions likely replicate at the same rate independent of RB numbers. Although these results demonstrate that *Chlamydia* replication rates are equivalent across inclusion, the true rate of replication remains obfuscated due to differing numbers of total $RB_E$s per inclusion. As chlamydial replication appears to be asymmetric late in the cycle, to determine the 'true' replication rate of *Chlamydia*, the average number of late cycle (>30 hpi) total $RB_E$s per inclusion would need to be determined and compared to late cycle EB production rate.

**Figure 5.1**
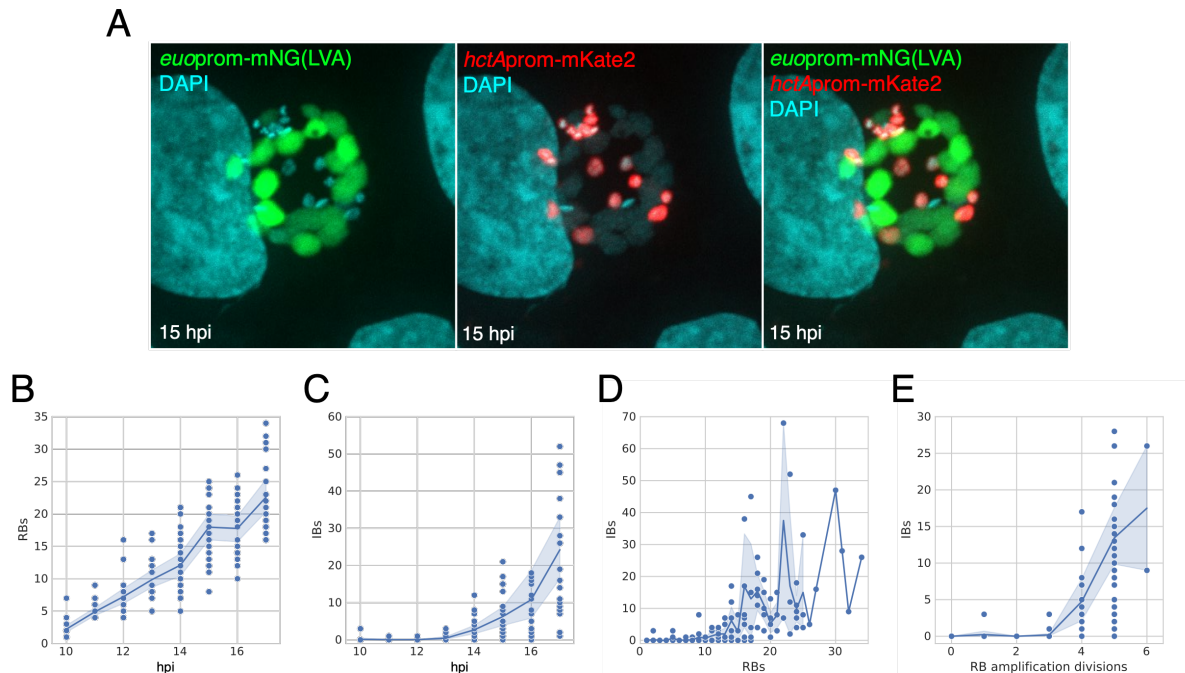


**Figure 5.1: Quantification of early RB and IB numbers.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-mNG(LVA)_*hctA*prom-mKate2 EBs. Samples were treated with penicillin for 4 hr from 10-17 hpi prior to fixation and staining with DAPI. (A) Representative confocal micrograph of individual *euo*prom-mNG(LVA)+: RBs, and *hctA*prom-mKate2+: IBs cells. Graphical representations of (B) the numbers of RBs per inclusion per time point, (C) the numbers of IBs per inclusion per time point, (D) the numbers of IBs per RBs per inclusion, and (E) the numbers of IBs per RBs per RB amplification divisions. Solid line represents the mean. Cloud represents 95% ci.

**Figure 5.2**
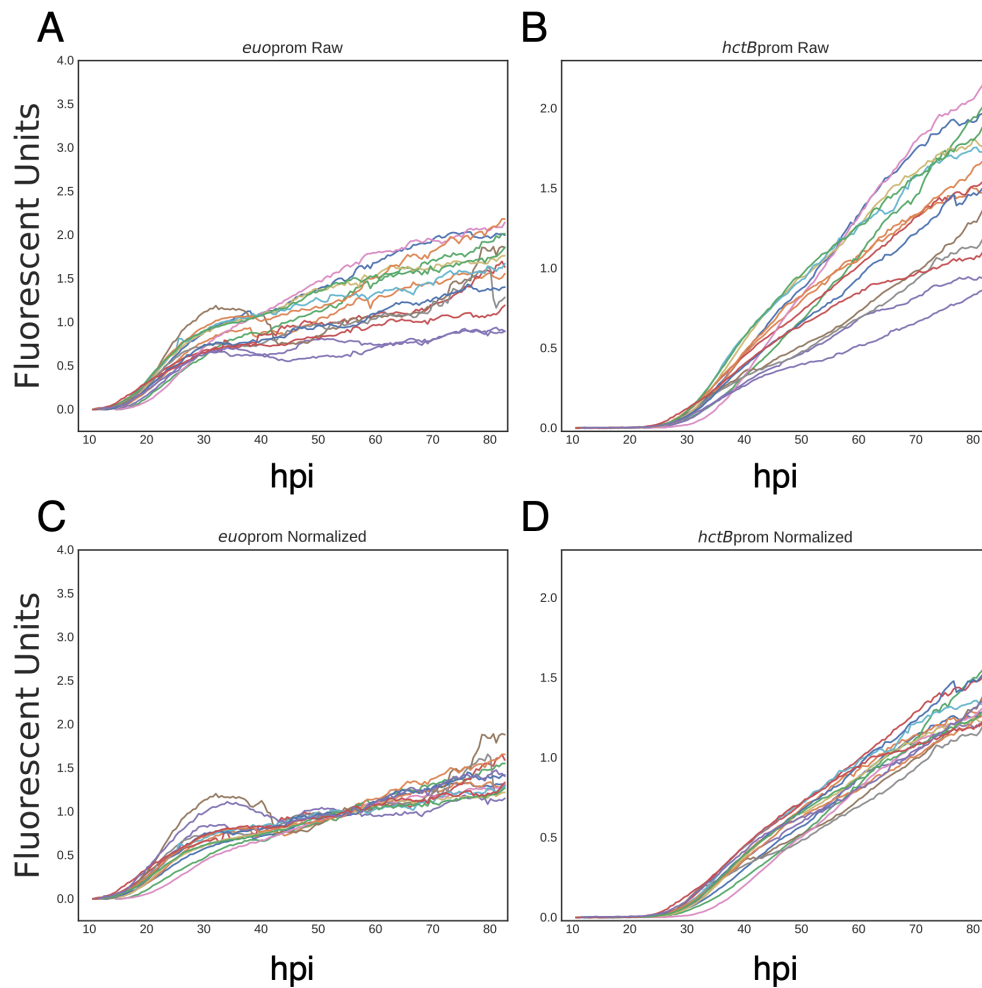


**Figure 5.2: Individual inclusion traces of *hctB*prom-mKate2 normalized to the level of respective *euo*prom-Clover reporter.**

Cos-7 cells were infected with purified *Ctr*-L2-*euo*prom-Clover_*hctB*prom-mKate2 EBs. Individual inclusions were imaged every 30 min from 10-80 hpi. (A-B) Live-cell fluorescent kinetic traces of individual inclusions expressing *euo*prom-Clover: RBs, and *hctB*prom-mKate2: EBs, respectively. (C-D) RB (*euo*prom-Clover) and EB-associated (*hctB*prom-mKate2) reporter kinetics normalized by the relative fluorescence from the plateau stage (50-60 hpi) of the paired RB-associated signal.

**Supplemental Material 5.S1: Normalization of EB Production**

Normalization_of_EB_production-Markdown

May 10, 2022

```
[ ]: %matplotlib notebook
     import matplotlib.pyplot as plt
     import matplotlib.ticker as ticker
     import numpy as np; np.random.seed(22)
     import seaborn as sns; sns.set(color_codes=True)
     import pandas as pd
     import math
```

```
[ ]: # Imports all csv files in folder and concatonates the data sets from trackmate

     import glob

     df = pd.DataFrame()
     for filename in glob.glob('data_raw/*.csv'):
         data_01 = pd.read_csv(filename, sep=',')
         filename = filename.split('/')[1]
         filename = filename.split('.')[0]
         well = filename.split('_')[6]
         data_01['Well'] = well.split('-')[0]
         data_01['FOV'] = filename.split('_')[7]
         data = data_01[['Frame','Track_ID', 'Well', 'FOV', 'X', 'Y', 'Channel_1',␣
      ↪'Channel_2']]
         df = df.append(data,  ignore_index=True)
     df['Sample_ID'] = df.Well + '-' + df.FOV
     df['Sample_ID'] = df.Sample_ID + '-Track-' + df.Track_ID.astype(str)
```

```
[ ]: # Subtract baseline new using min value: Channel_1.

     def subtract_bl(in_df):
         traces = in_df
         traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
      ↪values='Channel_1')
         df_test2 = pd.DataFrame()
         for columns in traces_p:
             minvalue = traces_p[columns].min()
             base_sub = lambda x: x-minvalue
             df_test = in_df[in_df['Sample_ID']==columns]
```

```
        df_test['bc_channel_2'] = df_test['Channel_1']-minvalue
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df)
```

```
[ ]: # Subtract baseline new using min value: Channel_2.

def subtract_bl(in_df):
    traces = in_df
    traces_p = traces.pivot_table(index='Frame', columns='Sample_ID',␣
  ↪values='Channel_2')
    df_test2 = pd.DataFrame()
    for columns in traces_p:
        minvalue = traces_p[columns].min()
        df_test = in_df[in_df['Sample_ID']==columns]
        df_test['bc_channel_1'] = df_test['Channel_2']-minvalue
        df_test2 = df_test2.append(df_test)
    return df_test2

df_bl = subtract_bl(df_bl)
```

```
[ ]: # Filtering out inclusion near the edges of the field of view.

df2 = df_f[~(df_f['X']<10)]
df2 = df2[~(df2['X']>670)]
df2 = df2[~(df2['Y']<10)]
df2 = df2[~(df2['Y']>670)]
df2 = df_f
```

```
[ ]: df2
```

```
[ ]: # Calibrate Frame values from the image slices to time values of experiment.

totalFrames = 150
startTime = 7.5
interval = 0.5

frame_dict = {}
for i in range(totalFrames):
    if i == 0:
        frame = i
        frame_dict[frame] = startTime+1
    else:
        frame = i
        startTime += interval
        frame_dict[frame] = startTime+1
```

```
df2['Time'] = df2['Frame'].map(frame_dict)
```

```
[ ]: # Save baseline subtracted data to pickle file.

     df2.to_pickle("baseLine_subtract_ssg.pkl")
```

```
[ ]: # Import baseline subtracted pickle file into Pandas dataframe.

     df2 = pd.read_pickle("baseLine_subtract_ssg.pkl")
```

```
[ ]: # Filter out traces that do not extend over two time points.

     df_f1 = df2['Sample_ID'][df2['Time']==30]
     df_f2 = df2['Sample_ID'][df2['Time']==75]
     df_f3 = df_f1[df_f1.isin(df_f2)]
     df3 = df2[df2['Sample_ID'].isin(df_f3)]
```

```
[ ]: # Assign treatments/strains to wells, HctB = euoprom-mNGLVA_hctBprom-mKate2.

     HctB = df3[df3['Sample_ID'].str.contains("A3").fillna(False)]
```

```
[ ]: # Filter for inclusions that exhibit sufficient expression in bc_channel_2.

     def filterI(in_df, threshold, time):
         filter_df=in_df[in_df['Time']==time]
         traces_p = filter_df.pivot_table(index='Time', columns='Sample_ID',␣
      ↪values='bc_channel_2')
         df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
         for columns in traces_p.columns:
             max_value = traces_p[columns].max()
             if max_value > threshold:
                 df_pass = df_pass.append({'Sample_ID':columns, 'pass': True},␣
      ↪ignore_index=True)

         new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
         return new_df

     HctB_f = filterI(HctB ,4000, 60)
```

```
[ ]: # Filter out inclusions that exhibit rupture in bc_channel_2.

     def filterII(in_df, threshold, time):
         filter_df=in_df[in_df['Time']==time]
         filter_df = in_df
         traces_p = filter_df.pivot_table(index='Time', columns='Sample_ID',␣
      ↪values='bc_channel_2')
         df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
```

3

```
      for columns in traces_p.columns:
          max_value = traces_p[columns].max()
          late_value = traces_p[columns].loc[time]
          if late_value > (max_value*0.5):
              df_pass = df_pass.append({'Sample_ID':columns, 'pass': True},␣
  ↪ignore_index=True)

      new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
      return new_df

HctB_f2 = filterII(HctB_f ,12, 70)
```

```
[ ]: # Filter for inclusions that exhibit sufficient expression in bc_channel_1.

def filterIII(in_df, threshold, time):
    filter_df=in_df[in_df['Time']==time]
    traces_p = filter_df.pivot_table(index='Time', columns='Sample_ID',␣
  ↪values='bc_channel_1')
    df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
    for columns in traces_p.columns:
        max_value = traces_p[columns].max()
        late_value = traces_p[columns].loc[time]
        if late_value > threshold:
            df_pass = df_pass.append({'Sample_ID':columns, 'pass': True},␣
  ↪ignore_index=True)

    new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
    return new_df

HctB_f3 = filterIII(HctB_f2 ,3000, 35)
```

```
[ ]: # Filter out outlier inclusion in bc_channel_1.

def filterIIII(in_df, threshold, time):
    filter_df=in_df[in_df['Time']==time]
    traces_p = filter_df.pivot_table(index='Time', columns='Sample_ID',␣
  ↪values='bc_channel_1')
    df_pass = pd.DataFrame(columns=['Sample_ID', 'pass'])
    for columns in traces_p.columns:
        max_value = traces_p[columns].max()
        late_value = traces_p[columns].loc[time]
        if late_value < threshold:
            df_pass = df_pass.append({'Sample_ID':columns, 'pass': True},␣
  ↪ignore_index=True)

    new_df = pd.merge(in_df, df_pass,  how='right', on=['Sample_ID'])
```

```
        return new_df

HctB_f4 = filterIIII(HctB_f3 ,40000, 80)
```

```
[ ]: # Normalize to # of RBs. Do this by normalizing to euo signal at 50-60 hpi␣
     ↪bc_channel_2.

     def subtract_bl(in_df, euo_time):
         traces = in_df
         traces_p1 = traces.pivot_table(index='Time', columns='Sample_ID',␣
     ↪values='bc_channel_1')
         traces_p2 = traces.pivot_table(index='Time', columns='Sample_ID',␣
     ↪values='bc_channel_2')
         df_test2 = pd.DataFrame()
         for columns in traces_p2:
             normvalue = traces_p1[columns].loc[euo_time:(euo_time+10)].mean()
             print(normvalue)
             df_test = in_df[in_df['Sample_ID']==columns]
             df_test['channel_2_norm'] = df_test['bc_channel_2']/normvalue
             df_test['channel_2_raw'] = df_test['bc_channel_2']/14416.0014
             df_test['norm_value'] = normvalue
             df_test2 = df_test2.append(df_test)
         return df_test2

     HctB_norm = subtract_bl(HctB_f4, 50)
```

```
[ ]: # Normalize to # of RBs. Do this by normalizing to euo signal at 50-60 hpi␣
     ↪bc_channel_1.

     def subtract_bl2(in_df, euo_time):
         traces = in_df
         traces_p1 = traces.pivot_table(index='Time', columns='Sample_ID',␣
     ↪values='bc_channel_1')
         traces_p2 = traces.pivot_table(index='Time', columns='Sample_ID',␣
     ↪values='bc_channel_2')
         df_test2 = pd.DataFrame()
         for columns in traces_p2:
             normvalue = traces_p1[columns].loc[euo_time:(euo_time+10)].mean()
             print(normvalue)
             df_test = in_df[in_df['Sample_ID']==columns]
             df_test['channel_1_norm'] = df_test['bc_channel_1']/normvalue
             df_test['channel_1_raw'] = df_test['bc_channel_1']/14416.0014
             df_test['norm_value'] = normvalue
             df_test2 = df_test2.append(df_test)
         return df_test2
```

```
HctB_norm_euo = subtract_bl2(HctB_f4, 50)
```

```
[ ]: # Save normalized data to .csv file.

Yes_Chem_norm.to_csv("EB_normalized_production_data.csv")
```

```
[ ]: # Pivot dataframe use Time as index and Sample_ID as columns.

def pivot(in_df, channel):
    in_df_p = in_df.pivot_table(index='Time', columns='Sample_ID',␣
 ↪values=channel)
    in_df_p['mean'], in_df_p['std'], in_df_p['HPI'] = in_df_p.mean(axis=1),␣
 ↪in_df_p.std(axis=1), in_df_p.index
    return in_df_p
```

```
[ ]: # Pivot dataframe

Euo_Yes_Chem_p   = pivot(HctB_norm_euo,  'channel_1_raw')
HcB_Yes_Chem_p   = pivot(HctB_norm,  'channel_2_raw')

Euo_Yes_Chem_n_p = pivot(HctB_norm_euo,  'channel_1_norm')
HcB_Yes_Chem_n_p = pivot(HctB_norm,  'channel_2_norm')
```

```
[ ]: # Plot individual inclusion traces.

from matplotlib.ticker import MultipleLocator
from matplotlib.ticker import AutoMinorLocator
from matplotlib.ticker import LogLocator
c = sns.color_palette('Set1',16).as_hex()
c[1]

with plt.style.context('seaborn-white'):
    fig, ((ax1,ax2),(ax3,ax4)) = plt.subplots(ncols=2, nrows=2)
    fig.tight_layout()


    (Euo_Yes_Chem_p).drop(columns=['std', 'mean','HPI']).plot(legend=False,␣
 ↪ax=ax1)
    HcB_Yes_Chem_p.drop(columns=['std', 'mean','HPI']).plot(legend=False,␣
 ↪ax=ax2)
    Euo_Yes_Chem_n_p.drop(columns=['std', 'mean','HPI']).plot(legend=False,␣
 ↪ax=ax3)
    HcB_Yes_Chem_n_p.drop(columns=['std', 'mean','HPI']).plot(legend=False,␣
 ↪ax=ax4)

    ax1.set_ylim([-0.25, 4])
```

```
ax2.set_ylim([-0.02, 2.3])
ax3.set_ylim([-0.25, 4])
ax4.set_ylim([-0.02, 2.3])

ax1.set_xlim([8, 83])
ax2.set_xlim([8, 83])
ax3.set_xlim([8, 83])
ax4.set_xlim([8, 83])


ax1.set_title('$euo$prom Raw', fontsize=14)
ax2.set_title('$hctB$prom Raw', fontsize=14)
ax3.set_title('$euo$prom Normalized', fontsize=14)
ax4.set_title('$hctB$prom Normalized', fontsize=14)
fig.set_size_inches(14, 12)

plt.savefig('euo_hctB_normalized.pdf')
```

CHAPTER SIX: DISCUSSION

**Summary of Findings**

*Chlamydia* undergoes a multi cell form developmental cycle consisting of both infectious and non-infectious cell forms. Completion of this cycle takes place within an intracellular vacuole, termed the inclusion, and is essential for chlamydial survival and proliferation. Over the last several decades, numerous studies have described both the cell forms and developmental dynamics of *Chlamydia* using an array of techniques (e.g. electron microscopy, transcriptomics, reinfection assays, and reporter strains). However, the mechanisms that control chlamydial cell-form development have yet to be elucidated. Uncovering these mechanisms would provide crucial targets for fighting this important pathogen.

**RB-to-EB development.** Dissemination of *Chlamydia* is reliant on the ability to transition from the reticulate body (RB) to the elementary body (EB). Without a known mechanism for RB-to-EB development, multiple regulatory hypotheses have been proposed [1, 2, 3, 4]. In **Chapter 2** we divided these developmental hypotheses into two overarching categories, environmentally or intrinsically derived signals. In this study we showed that the timing of RB-to-EB development was unaffected by external competition (either increased MOI or superinfection) (**Fig. 2.4, 2.5**). Chlamydial growth and cell-form development, however, were dependent on fluctuations of the infectious environment temperature (**Fig. 2.3, 2.S4**) and on inhibition of host protein synthesis (cycloheximide treatment) (**Fig. 2.10**). No detectable expression from the RB or EB-associated promoters was observed when the infectious environment was depleted of either iron or tryptophan (bipyridyl and IFN-γ treated, respectively) (**Fig. 2.S2**), indicating that both chlamydial growth and cell-form development were inhibited. Furthermore, IB and EB development were both prevented when cell division was inhibited with penicillin-G or D-cycloserine (**Fig. 2.6**). These results indicated that RB-to-EB differentiation follows an intrinsic developmental program dependent on bacterial growth and cell division. *Chlamydia's* inability to respond to external stimuli in regards to RB-to-EB

differentiation differs from other intracellular bacteria (e.g. *Coxiella, Legionella,* and *Mycobacterium*) which utilize components of the stringent response pathway (RelA, SpoT, and RpoS) to regulate their gene expression [5, 6, 7, 8]. However, these results are consistent with previous genomic studies which showed that *Chlamydia* does not encode *relA, spoT,* or *rpoS* [9, 10].

Results from **Chapter 2** indicated that RB-to-EB development was dependent on cell division. Therefore, to investigate the relationship between cell division and RB-to-EB development we created three alternative dual promoter-reporter chlamydial strains to monitor active cell-form specific expression at the single-cell level. Analysis of these strains revealed that active RB, IB, and EB-specific expression was isolated to individual cells (**Fig. 3.1**). Live-cell kinetics from the *euo*prom reporter strain demonstrated similar results to those observed in **Chapter 2** (an increase in expression from ~12 to 24 hpi, followed by a late cycle plateau). Kinetics from the *hctA*prom reporter demonstrated initiation of expression at ~18 hpi also followed by a plateau 10 hours later (**Fig. 3.1**). We confirmed that the increase and plateau in *euo* and *hctA*prom inclusion-level expression corresponded to RB and IB cell numbers within individual inclusions (**Fig. 3.2**). Using this data we created agent-based models to distinguish between two possible mechanisms of IB development: asymmetric production of IBs from $RB_E$s and direct conversion of $RB_E$s into IBs (**Fig. 3.3**). The results from live-cell analysis of intra-inclusion RB populations and the dynamics of individual RBs were consistent with the simulated outputs from the asymmetric production model, where late in the infectious cycle (>24 hpi), the number of RBs was unchanged and individual RBs were static (**Fig. 3.5, 3.6**). Also in agreement with the simulated output from the asymmetric production model was the individual inclusion expression kinetics of *hctB*prom which, after 36 hpi, continued on a near perfect linear trajectory.

Our models predicted that we would be able to further distinguish between the developmental hypotheses by inhibiting chlamydial cell division. Cell division inhibition experiments (penicillin and ciprofloxacin treated infections) showed that the RB population was again unchanged and that further IB production was inhibited (**Fig. 3.8, 3.10**). Together, these results indicated that development of the IB was a

cell division dependent process, likely occurring by asymmetric production from the RB.

The involvement of asymmetric division in differential cell-form development has been documented in both prokaryotes and eukaryotes [11, 12]. The two of the most well-studied bacterial examples of asymmetric division are mother-to-endospore production in *Bacillus subtilis* and stalked-to-swarmer production in *Caulobacter crescentus* [11]. In *C. crescentus*, both the stalk (in stalked cells) and the flagella (in swarmer cells) are polarly localized. An analogous architecture is seen in *Chlamydia*, as both RBs and EBs have been shown to have asymmetrical cell topology, exhibiting Type III secretion systems at a single cell pole in electron micrographs and by immunofluorescence [13, 14]. Chlamydial cell division has also appears to progress asymmetrically, as the peptidoglycan septum ring was shown to be produced at a single pole rather than equatorially and asymmetric protein expression occurred on either side of the division plane in newly budding *Chlamydia* [15, 16]. The culmination of work from this dissertation as well as the aforementioned studies provide substantial evidence that supports *Chlamydia's* use of asymmetric division in the formation of IBs.

**IB-to-EB development.** Throughout this study we found that the *hctA* promoter-reporter exhibited alternative expression patterns when compared to the other late gene promoters (*hctB*, *scc2*, and *tarp*). We first found that native expression from the *hctA* promoter initiated earlier in the developmental cycle at ~18-20 hpi for *hctA* versus ~24 hpi for *hctB*, *scc2*, and *tarp* (**Fig. 2.4, 2.5, 2.6, 2.S1**). When chlamydial cell division was inhibited (treated with penicillin-G and D-cycloserine) expression from the *hctA* promoter-reporter immediately halted, followed by re-initiation in aberrant cells ~10 hour post treatment (**Fig. 2.6, 2.7, 2.8, 2.9**). This differed from *hctB*, which continued linearly, similar to the untreated sample, for ~10 hours post treatment, before abruptly plateauing (**Fig. 2.8**). *hctB* expression also only occurred in small EB-like cells (**Fig. 2.9**). When analyzing single-cell reporter localization using our active promoter-reporters in **Chapter 3**, we also found that active *hctA* promoter expression existed in separate cells from those that expressed *hctB* (**Fig.**

**3.1**). The differences in the timing, regulation, and cell-form localization of the *hctA* and *hctB* promoter-reporters led us to reclassify *hctA* as an intermediate body (IB)-associated gene.

The alternative expression patterns of *hctA* and *hctB* in cell division-inhibited *Chlamydia* from **Chapter 2** suggested to us that IB-to-EB development is a committed step, independent of cell division; and that IB-to-EB maturation takes ~10 hours to occur. In **Chapter 3**, using this assumption, we modeled IB-to-EB development as occurring by direct maturation (**Fig. 3.3**). The outputs of this model predicted that the IB population would be at steady-state and individual IBs would be transient as they were produced by RBs and then converted into EBs (**Mov. 3.S1, 3.S2**). In agreement with the IB-to-EB direct maturation hypothesis, individual IBs (as measured by our active *hctA* promoter-reporter) appeared and disappeared throughout the infectious cycle (**Fig. 3.6** and **Mov. 3.S4**). *hctA* promoter activity was also extremely brief in individual IBs, lasting for only 0.5-1 hour. IB-to-EB direct maturation was also confirmed by development of small EB-like (*hctB*prom positive) cells and the overlap in IB and EB reporter expression within single cells after cell division inhibition or RB cell lysis (**Fig. 3.11, 3.12, 3.13**).

HctA, due to its expression late in the developmental cycle and its ability to bind and condense DNA at high levels in *Escherichia coli*, producing a similar phenotype to that of the condensed EB nucleoid, has classically been considered an EB-associated protein. However, due to HctA's the ability to induced changes in DNA topology *in vitro* and modulate changes in gene expression within *E. coli* when expressed at substructural levels, it has also been hypothesized that HctA is an H-NS-like protein capable of regulating genes involved in EB development [17, 18, 19]. In our study, both the timing of *hctA* expression (~18 hpi) and the brief promoter activity of *hctA* in individual cells followed by the expression of late-stage EB genes are in agreement with this hypothesis and suggest that IB-to-EB development may be dependent on a regulatory cascade. Further evidence of the involvement of a regulatory cascade in IB-to-EB maturation is the CtcB/CtcC two component regulatory system. *ctcB* and *ctcC* have been shown to be expressed at 18 hpi, corresponding to the initial time of IB-to-EB development, and over-expression of

CtcC in *Chlamydia*, showed that CtcC is upstream of $\sigma^{54}$ and positively regulates a large subset of $\sigma^{54}$-dependent EB-associated genes [20]. The results from this dissertation and other studies strongly suggest that, after asymmetric production of the IB, IB-to-EB development occurs by direct maturation and utilizes multiple regulatory cascades.

**$RB_R$ and $RB_E$ subpopulations.** The majority of the results presented in **Chapter 2** utilized aggregate data (including inclusions that had lysed as well as reinfections). However, analysis of well-isolated inclusions revealed that each inclusion followed similar fluorescent dynamics, each consisting of two primary stages. During the first stage, individual inclusions demonstrated an increase in *euo* reporter expression from ~12 to 24 hpi, this was followed by a plateau in the *euo* reporter which corresponded to a linear increase from *hctB*prom-mKate2 until host cell lysis (**Fig. 2.10, Mov. 2.S2**). Our interpretation of this data was that RBs were undergoing a maturation process, separating the RB population into two subpopulations: immature RBs, or $RB_R$s, which divide symmetrically to amplify the total RB population (~12 to 24 hpi), and mature RBs, or $RB_E$s, which undergo asymmetric division to linearly produce IBs (and subsequently EBs) from ~24 hpi until host cell lysis.

As $RB_R$-to-$RB_E$ maturation is a newly described phenomena in *Chlamydia*, we sought to better describe the dynamics-of and determine the potential factors involved in $RB_R$-to-$RB_E$ maturation by quantifying the number of RBs and IBs early in the developmental cycle (10-17 hpi). Preliminary single-cell RB and IB counts demonstrated that IBs were present as early as the first RB division and continued throughout the RB amplification period (**Fig. 5.1**). These results suggested that both immature $RB_R$s and mature $RB_E$s exist simultaneously and that $RB_R$-to-$RB_E$ maturation was not dependent on RB cell division or RB number.

Isolated live-cell inclusion traces from **Chapters 2** suggested that the maximum number of $RB_E$s corresponded directly to the rate of EB production (**Mov. 2.S2**). By normalizing the expression slopes of *hctB*prom-mKate2 (EBs) from individual inclusions by their respective *euo*prom-Clover (RBs) levels, our preliminary results suggest that chlamydial replication rates are similar between inclusions and that the

rate of EB production is in fact dependent on the total number of $RB_E$s during the plateau period (**Fig. 5.2**). These results also suggest that the differences seen in the individual inclusion RB plateau levels and EB production rates are due to the rate of $RB_R$-to-$RB_E$ maturation and not the replication rates of individual intra-inclusion populations.

*Chlamydia's* ability to first proceed through an amplification stage ($RB_R$s) then switch to a stem cell population ($RB_E$s) is a newly described phenomenon in bacteria. The closest examples that resemble this are in *Vibrio cholerae*, *Myxococcus xanthus*, and *B. subtilis*, which all undergo vegetative growth followed by either the formation or production of disseminating or environmentally stable cell forms [21, 22, 23]. However, these bacteria utilize environmental signals (e.g. quorum sensing and nutrient starvation) to initiate these processes, whereas our current data suggests that *Chlamydia* does not [21, 22, 24]. Uncovering the mechanisms that control this critical step in *Chlamydia* is an important future direction of research.

**Live-cell chlamydial mutagenesis.** The research in this dissertation has provided unprecedented detail in regards to chlamydial cell-form development. **Figure 6.1** is a schematic representation of our current interpretation of the developmental cycle including each cell-form specific stage. Many of the mechanisms from this study were, however, described at a higher level (i.e. intrinsic RB-to-EB differentiation signal, $RB_E$/IB asymmetric division, and IB-to-EB direct maturation). Therefore, to uncover the genes involved in chlamydial cell-form development, in **Chapter 4,** we established an automated live-cell mutagenesis protocol to isolate mutants exhibiting defects in the developmental cycle using one of our cell-form specific fluorescent reporter strains. Purified EBs from this strain were mutagenized directly in axenic media (CIP-1) containing the known chemical mutagen ethyl methanesulfonate [1]. The mutagenized *Chlamydia* were then monitored by live-cell fluorescence microscopy, and *Chlamydia* exhibiting altered developmental profiles were isolated from the population by micromanipulation. Developmental abnormalities in isolated mutagenized strains were then verified by a subsequent round of infection and live-cell microscopy. Future use of this technique

will be a powerful tool in identifying the genes involved in each stage of cell-form development in *Chlamydia*.

**Future directions**

   **Transcriptional profiling of cell-form subpopulations.** Previous temporal microarray and RNA sequencing assays have revealed stage-specific transcriptional profiles throughout the infectious cycle [25, 26]. As we demonstrated, the developmental cycle potentially consists of four distinct cell-form subpopulations: $RB_R$s, $RB_E$s, IBs, and EBs.   However, many of these cell forms coexist temporally within individual inclusions, confounding population-level transcriptomic data [27]. Although bacterial single-cell RNA sequencing has lagged in comparison to eukaryotic assays, there has been recent success in sequencing the transcripts of individual bacteria from mixed populations of *E. coli* and *B. subtilis* [28]. Single-cell bacterial RNA sequencing uses a modified version of Split Pool Ligation-based Transcriptome sequencing (SPLiT-seq) to produce and tag the cDNA within individual bacterium with unique "barcodes", therefore circumventing the need to isolate individual subpopulations [28, 29]. Of note, Kuchina et al. 2021, implemented this method to determine the transcriptional profiles of individual cell forms in *B. subtilis* under various stress inducing conditions [28]. Assaying stage-specific time points with SPLiT-seq would be a powerful tool to separate out the transcriptional profiles of each chlamydial subpopulation, providing a more precise list of cell-form specific regulated genes. With the advent of novel inducible control (aTC and riboswitch) and conditional knock-down (CRISPRi) systems, targeted approaches could be used to investigate genes differentially regulated within each cell form to determine their impacts on development [30, 31].

   **Identifying genes involved in cell-form development.** We have broken down cell-form development into several stages: 1. EB-to-RB germination, 2. $RB_R$ amplification, 3. $RB_R$-to-$RB_E$ maturation, 4. Asymmetric IB production, and 5. IB-to-EB direct maturation (**Fig. 6.1**). However, the mechanisms that control each step are unknown. Uncovering the genes that regulate each process is therefore an important

avenue of research. In **Chapter 4,** using live-cell microscopy and a chlamydial cell-form specific reporter**,** we established a mutagenesis protocol to isolate mutants that exhibited defects in the developmental cycle. Modifying this system to include multiple stage-specific reporter strains would allow for the discovery of genes involved in each step of development.

A major issue with this technique is that cell-form development is essential for chlamydial dissemination, therefore creating mutations in these pathways will likely lead to non-viable *Chlamydia*. To circumvent this problem, we proposed that after isolation of *Chlamydia* from individual inclusion, whole-genome sequencing could be performed without replating to identify the genetic lesions associated with non-viable isolated strains. Once these mutations have been identified, each mutation could then be tested by the creation of individual conditional knockdowns using the newly developed CRISPRi system [31]. Cell-form development is essential for chlamydial proliferation and infection; therefore, the genes that control these processes should be elucidated.

**Determining the mechanisms of asymmetric division.** Production of the IB is essential for completion of the infectious cycle, yet the mechanisms of IB development are unknown. In **Chapter 3**, we showed that further development of the IB is dependent on cell division from RBs by inhibiting chlamydial cell division and inducing lysis of the RB population (**Fig. 3.10**). Although these results strongly implicated asymmetric division as the likely mechanism for IB production, we were unable to produce direct evidence of asymmetric division with our current reporter strains (e.g. single-cell confocal micrographs of asymmetrically dividing cells). Capturing $RB_E$/IB asymmetrical division would be a large advancement in our knowledge of chlamydial cell-form development and an obvious future direction. Several studies have shown asymmetric budding of *Chlamydia* using EDA-DA, a novel peptidoglycan tagging technique, as well as polar protein localization in pre-divisional cells [16, 32]. However, neither process was linked specifically to cell-form development. Using single-cell RNA sequencing (mentioned in the previous section), $RB_E$/IB specific genes could be determined and used to develop alternative

chlamydial reporters. Combining these reporters with the EDA-DA peptidoglycan tagging system, would allow for visualization of asymmetric reporter expression on either side of the division plane. An alternative approach would be to fuse an epitope tag or fluorescent protein to $RB_E$/IB specifically localized proteins, similarly to what has been done in *Caulobacter crescentus* and *B. subtilis* [33, 34].

IB, and therefore EB, production is reliant on cell division from the RB, therefore fully understanding both the genetic and physical mechanisms involved in this process is important. *Chlamydia* encodes an array of regulatory components utilized by other bacteria to alter their transcriptional profiles [35, 36, 37]. Of note, are the two chlamydial alternative sigma factors, $\sigma^{28}$ and $\sigma^{54}$. $\sigma^{28}$ is used across bacteria to control a range of processes including flagellar synthesis and endospore formation [38, 39]. In *Chlamydia*, $\sigma^{28}$ expression begins at ~12 hpi and continues throughout the developmental cycle [40]. Yu et al. showed that $\sigma^{28}$ regulates a subset of late EB-associated genes, including *hctB* and *tsp*, *in vitro* [41]. As minimum research has been done in regards to $\sigma^{28}$ in *Chlamydia*, neither the $\sigma^{28}$ regulon nor the mechanisms that control $\sigma^{28}$ expression have been determined. $\sigma^{54}$ is also present in numerous bacterial species and regulates an array of processes from cell-form differentiation to nitrogen fixation [36]. In *Chlamydia*, Soules et al. demonstrated that a large set of EB-associated genes, including 28 Type III secreted effectors, were controlled by $\sigma^{54}$. They further showed that the $\sigma^{54}$ regulon was downstream of/and controlled by the chlamydial two component regulatory system, CtcB/CtcC [20]. Although the CtcB/CtcC-$\sigma^{54}$ pathway has been elucidated, the upstream mechanisms that control CtcB/CtcC in *Chlamydia* are not known. Lastly, *Chlamydia* encodes multiple *clp* (caseinolytic protease) genes, including *clpC*, *clpX*, *clpP1*, and *clpP2*. The Clp protease system is commonly used to regulate shifts in protein expression by active protein degradation [42]. Wood et al. 2018, showed that expression of all four genes peaked at 16 hpi, a time concurrent with formation of the IB, however protein detection did not occur until 24 hpi, corresponding to IB-to-EB maturation [43]. Overexpression of non-viable *clpX* and *clpP2* mutants were also shown to negatively impact EB production, suggesting that these systems are used in RB-to-EB differentiation [44]. Again, the regulatory mechanisms that control *clp* gene and protein

expression have not been identified nor have the Clp-specific protein targets. Further investigation of the above pathways will expand our knowledge of the mechanisms that *Chlamydia* employs in asymmetric division and cell-form development.

**Future modeling of chlamydial development.** This dissertation has relied heavily on computational modeling to determine the mechanisms utilized in chlamydial cell-form development. In **Chapter 2,** we implemented systems of ordinary differential equations to distinguish between developmental hypotheses and determine the nature of the signal for RB-to-EB differentiation. Whereas, in **Chapter 3**, we constructed agent-based models to explore the relationship between development of the IB and cell division. The major advantage of computational modeling in both studies was the ability to simulate multiple conditions *in silico* to guide *in vivo* experiments. Although our most current model is capable of simulating the overall kinetics of chlamydial development, many of the cell-form specific dynamics (i.e. EB-to-RB germination, $RB_R$ amplification, and $RB_R$-to-$RB_E$ maturation) were modeled after empirical data as the biological mechanisms have yet to be elucidated. Therefore, to uncover the mechanisms that control each of these stages, further implementation and modification of our computational models should be continued.

**Final Thoughts.** This dissertation has demonstrated the strength of utilizing cell-form specific reporter strains and automated live-cell microscopy to monitor chlamydial development dynamics during active infections. We revealed that chlamydial cell-form development is a complex multi-step process, consisting of four potential cell-form subpopulations. Although we described many of these processes at a higher-level (i.e. $RB_R$-to-$RB_E$ maturation, $RB_E$/IB asymmetric division, and IB-to-EB direct maturation), how each stage of development is regulated remains unclear. Continued implementation of our *in silico* models as well as our automated live-cell promoter-reporter system and mutagenesis protocol will support future research in uncovering the mechanisms and genetic components involved in chlamydial cell-form development.

**References**

1. Omsland A, Sager J, Nair V, Sturdevant DE, Hackstadt T. Developmental stage-specific metabolic and transcriptional activity of Chlamydia trachomatis in an axenic medium. *Proceedings of the National Academy of Sciences of the United States of America*. 2012;109(48):19781-19785. doi:10.1073/pnas.1212831109

2. Lee JK, Enciso GA, Boassa D, Chander CN, Lou TH, Pairawan SS, Guo MC, Wan FYM, Ellisman MH, Stterlin C, Tan M. Replication-dependent size reduction precedes differentiation in Chlamydia trachomatis. *Nature communications*. 2018;9(1). doi:10.1038/s41467-017-02432-0

3. Peters J, Wilson DP, Myers G, Timms P, Bavoil PM. Type III secretion  la Chlamydia. *Trends in microbiology*. 2007;15(6). doi:10.1016/j.tim.2007.04.005

4. Hoare A, Timms P, Bavoil PM, Wilson DP. Spatial constraints within the chlamydial host cell inclusion predict interrupted development and persistence. *BMC microbiology*. 2008;8. doi:10.1186/1471-2180-8-5

5. Moormeier DE, Sandoz KM, Beare PA, Sturdevant DE, Nair V, Cockrell DC, Miller HE, Heinzen RA. Coxiella burnetii RpoS Regulates Genes Involved in Morphological Differentiation and Intracellular Growth. *Journal of bacteriology*. 2019;201(8). doi:10.1128/JB.00009-19

6. Dalebroux ZD, Edwards RL, Swanson MS. SpoT governs Legionella pneumophila differentiation in host macrophages. *Molecular microbiology*. 2009;71(3). doi:10.1111/j.1365-2958.2008.06555.x

7. Bachman MA, Swanson MS. RpoS co-operates with other factors to induce Legionella pneumophila virulence in the stationary phase. *Molecular Microbiology*. 2000;40(5):1201-1214. doi:https://doi.org/10.1046/j.1365-2958.2001.02465.x

8. Primm TP, Andersen SJ, Mizrahi V, Avarbock D, Rubin H, 3rd CEB. The stringent response of Mycobacterium tuberculosis is required for long-term survival. *Journal of Bacteriology*. 2000;182(17). doi:10.1128/jb.182.17.4889-4898.2000

9. Kalman S, Mitchell W, Marathe R, Lammel C, Fan J, Hyman RW, Olinger L, Grimwood J, Davis RW, Stephens RS. Comparative genomes of Chlamydia

pneumoniae and C. trachomatis. *Nature genetics*. 1999;21(4). doi:10.1038/7716

10. Mittenhuber G. Comparative genomics and evolution of genes encoding bacterial (p)ppGpp synthetases/hydrolases (the Rel, RelA and SpoT proteins). *Journal of molecular microbiology and biotechnology*. 2001;3(4).

11. Shapiro L, McAdams HH, Losick R. Generating and exploiting polarity in bacteria. *Science (New York, NY)*. 2002;298(5600). doi:10.1126/science.1072163

12. Sunchu B, Cabernard C. Principles and mechanisms of asymmetric cell division. *Development (Cambridge, England)*. 2020;147(13). doi:10.1242/dev.167650

13. Nans A, Saibil HR, Hayward RD. Pathogenhost reorganization during Chlamydia invasion revealed by cryo-electron tomography. *Cellular Microbiology*. 2014;16(10). doi:10.1111/cmi.12310

14. Dumoux M, Nans A, Saibil HR, Hayward RD. Making connections: snapshots of chlamydial type III secretion systems in contact with host membranes. *Current opinion in microbiology*. 2015;23. doi:10.1016/j.mib.2014.09.019

15. Cox JV, Abdelrahman YM, Ouellette SP. Penicillin-binding proteins regulate multiple steps in the polarized cell division process of Chlamydia. *Scientific reports*. 2020;10(1). doi:10.1038/s41598-020-69397-x

16. Abdelrahman Y, Ouellette SP, Belland RJ, Cox JV. Polarized Cell Division of Chlamydia trachomatis. *PLoS Pathogens*. 2016;12(8). doi:10.1371/journal.ppat.1005822

17. Shaw EI, Dooley CA, Fischer ER, Scidmore MA, Fields KA, Hackstadt T. Three temporal classes of gene expression during the Chlamydia trachomatis developmental cycle. *Molecular microbiology*. 2000;37(4):913-925. doi:https://doi.org/10.1046/j.1365-2958.2000.02057.x

18. Barry CE, Hayes SF, Hackstadt T. Nucleoid condensation in Escherichia coli that express a chlamydial histone homolog. *Science (New York, NY)*. 1992;256(5055):377-379. doi:10.1126/science.256.5055.377

19. Barry CE, Brickman TJ, Hackstadt T. Hc1-mediated effects on DNA structure: a potential regulator of chlamydial development. *Molecular microbiology*. 1993;9(2):273-283. doi:10.1111/j.1365-2958.1993.tb01689.x

20. Soules KR, LaBrie SD, May BH, Hefty PS. Sigma 54-Regulated Transcription Is Associated with Membrane Reorganization and Type III Secretion Effectors during Conversion to Infectious Forms of Chlamydia trachomatis. *mBio*. 2020;11(5). doi:10.1128/mBio.01725-20

21. Bridges AA, Fei C, Bassler BL. Identification of signaling pathways, matrix-digestion enzymes, and motility components controlling Vibrio cholerae biofilm dispersal. *Proceedings of the National Academy of Sciences of the United States of America*. 2020;117(51). doi:10.1073/pnas.2021166117

22. Kroos L. The Bacillus and Myxococcus Developmental Networks and Their Transcriptional Regulators. *Annual Review of Genetics*. 2007;41(1):13-39. doi:10.1146/annurev.genet.41.110306.130400

23. Levin PA, Grossman AD. Cell cycle and sporulation in Bacillus subtilis. *Current opinion in microbiology*. 1998;1(6). doi:10.1016/s1369-5274(98)80107-0

24. Phillips ZEV, Strauch MA. Bacillus subtilis sporulation and stationary phase gene expression. *Cellular and molecular life sciences: CMLS*. 2002;59(3). doi:10.1007/s00018-002-8431-9

25. Belland RJ, Zhong G, Crane DD, Hogan D, Sturdevant D, Sharma J, Beatty WL, Caldwell HD. Genomic transcriptional profiling of the developmental cycle of Chlamydia trachomatis. *Proceedings of the National Academy of Sciences of the United States of America*. 2003;100(14):8478-8483. doi:10.1073/pnas.1331135100

26. Grieshaber S, Grieshaber N, Yang H, Baxter B, Hackstadt T, Omsland A. Impact of Active Metabolism on Chlamydia trachomatis Elementary Body Transcript Profile and Infectivity. *Journal of bacteriology*. 2018;200(14). doi:10.1128/JB.00065-18

27. Friis RR. Interaction of L cells and Chlamydia psittaci: entry of the parasite and host responses to its development. *Journal of bacteriology*. 1972;110(2). doi:10.1128/jb.110.2.706-721.1972

28. Kuchina A, Brettner LM, Paleologu L, Roco CM, Rosenberg AB, Carignano A, Kibler R, Hirano M, DePaolo RW, Seelig G. Microbial single-cell RNA sequencing by split-pool barcoding. *Science (New York, NY)*. 2021;371(6531).

doi:10.1126/science.aba5257

29. Rosenberg AB, Roco CM, Muscat RA, Kuchina A, Sample P, Yao Z, Graybuck LT, Peeler DJ, Mukherjee S, Chen W, Pun SH, Sellers DL, Tasic B, Seelig G. Single-cell profiling of the developing mouse brain and spinal cord with split-pool barcoding. *Science (New York, NY)*. 2018;360(6385). doi:10.1126/science.aam8999

30. Grieshaber NA, Chiarelli TJ, Appa CR, Neiswanger G, Peretti K, Grieshaber SS. Translational gene expression control in Chlamydia trachomatis. *PloS one*. 2022;17(1). doi:10.1371/journal.pone.0257259

31. Ouellette SP, Blay EA, Hatch ND, Fisher-Marvin LA. CRISPR Interference To Inducibly Repress Gene Expression in Chlamydia trachomatis. *Infection and Immunity*. 2021;89(7). doi:10.1128/IAI.00108-21

32. Liechti G, Kuru E, Packiam M, Hsu YP, Tekkam S, Hall E, Rittichier JT, VanNieuwenhze M, Brun YV, Maurelli AT. Pathogenic Chlamydia Lack a Classical Sacculus but Synthesize a Narrow, Mid-cell Peptidoglycan Ring, Regulated by MreB, for Cell Division. *PLoS pathogens*. 2016;12(5). doi:10.1371/journal.ppat.1005590

33. Hinz AJ, Larson DE, Smith CS, Brun YV. The Caulobacter crescentus polar organelle development protein PodJ is differentially localized and is required for polar targeting of the PleC development regulator. *Molecular microbiology*. 2003;47(4). doi:10.1046/j.1365-2958.2003.03349.x

34. Rudner DZ, Pan Q, Losick RM. Evidence that subcellular localization of a bacterial membrane protein is achieved by diffusion and capture. *Proceedings of the National Academy of Sciences of the United States of America*. 2002;99(13). doi:10.1073/pnas.132235899

35. Ishihama A. Building a complete image of genome regulation in the model organism Escherichia coli. *The Journal of general and applied microbiology*. 2018;63(6). doi:10.2323/jgam.2017.01.002

36. Kustu S, Santero E, Keener J, Popham D, Weiss D. Expression of sigma 54 (ntrA)-dependent genes is probably united by a common mechanism. *Microbiological reviews*. 1989;53(3). doi:10.1128/mr.53.3.367-376.1989

37. Mashruwala AA, Eilers BJ, Fuchs AL, Norambuena J, Earle CA, van de Guchte A, Tripet BP, Copi V, Boyd JM. The ClpCP Complex Modulates Respiratory Metabolism in Staphylococcus aureus and Is Regulated in a SrrAB-Dependent Manner. *Journal of bacteriology*. 2019;201(15). doi:10.1128/JB.00188-19

38. Helmann JD. Alternative sigma factors and the regulation of flagellar gene expression. *Molecular microbiology*. 1991;5(12). doi:10.1111/j.1365-2958.1991.tb01847.x

39. Fimlaid KA, Shen A. Diverse mechanisms regulate sporulation sigma factor activity in the Firmicutes. *Current opinion in microbiology*. 2015;24. doi:10.1016/j.mib.2015.01.006

40. Douglas AL, Hatch TP. Expression of the transcripts of the sigma factors and putative sigma factor regulators of Chlamydia trachomatis L2. *Gene*. 2000;247(1-2). doi:10.1016/S0378-1119(00)00094-9

41. Yu HHY, Kibler D, Tan M. In silico prediction and functional validation of sigma28-regulated genes in Chlamydia and Escherichia coli. *Journal of bacteriology*. 2006;188(23). doi:10.1128/JB.01082-06

42. Pan Q, Losick R. Unique degradation signal for ClpCP in Bacillus subtilis. *Journal of bacteriology*. 2003;185(17). doi:10.1128/JB.185.17.5275-5278.2003

43. Wood NA, Chung KY, Blocker AM, de Almeida NR, Conda-Sheridan M, Fisher DJ, Ouellette SP. Initial Characterization of the Two ClpP Paralogs of Chlamydia trachomatis Suggests Unique Functionality for Each. *Journal of bacteriology*. 2018;201(2). doi:10.1128/JB.00635-18

44. Wood NA, Blocker AM, Seleem MA, Conda-Sheridan M, Fisher DJ, Ouellette SP. The ClpX and ClpP2 Orthologs of Chlamydia trachomatis Perform Discrete and Essential Functions in Organism Growth and Development. *mBio*. 2020;11(5). doi:10.1128/mBio.02016-20

**Figure 6.1.**
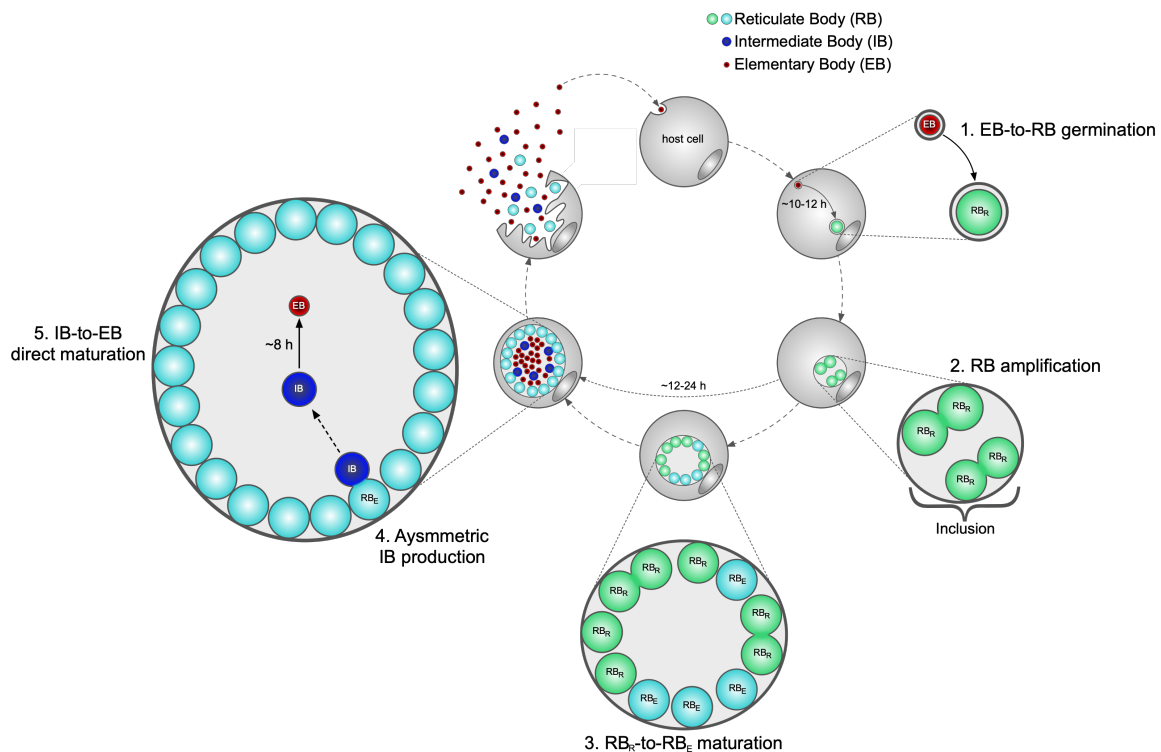


**Figure 6.1: Schematic of cell-form development throughout the infectious cycle.**

RBs; $RB_R$: green, $RB_E$: cyan. IB: dark blue. EB: red. After internalization, cell type development is divided into 5 distinct stages. **1. EB-to-RB germination** is a process that takes between 10-12 h, where the initial EB matures into the replication-competent RB. **2. RB amplification** is an increase in RB numbers due to the symmetric division of the $RB_R$ subpopulation. **3. $RB_R$-to-$RB_E$ maturation** is the conversion of $RB_R$s into $RB_E$s; $RB_E$s are a subset of RBs that are capable of asymmetric division and IB production. RB amplification and $RB_R$-to-$RB_E$ maturation appear to occur simultaneously, between approximately 12-24 hpi. After 24 hpi, the entire RB population appears to consist of $RB_E$s **4. IB asymmetric production**, the $RB_E$ is a stem cell-like (mother) cell type that divides asymmetrically to produce an IB daughter cell upon each division event. **5. IB-to-EB direct maturation** is cell division independent and takes approximately 8 h to occur.