7

Coordinated Action of N-CAM, N-cadherin, EphA4, and ephrinB2 Translates Genetic Prepatterns into Structure during Somitogenesis in Chick

James A. Glazier,[*,†] *Ying Zhang,*[*] *Maciej Swat,*[*] *Benjamin Zaitlen,*[*] *and Santiago Schnell*[†,*]

[*]Biocomplexity Institute and Department of Physics, 727 East Third Street,
Indiana University, Bloomington, Indiana 47405
[†]Indiana University School of Informatics, 1900 East Tenth Street, Indiana University,
Bloomington, Indiana 47406

Current Topics in Developmental Biology, Vol. 81
0070-2153/08 $35.00
DOI: 10.1016/S0070-2153(07)81007-6

During gastrulation in vertebrates, mesenchymal cells at the anterior end of the *presomitic mesoderm* (PSM) periodically compact, transiently epithelialize and detach from the posterior PSM to form somites. In the prevailing *clock-and-wavefront* model of somitogenesis, periodic gene expression, particularly of Notch and Wnt, interacts with an FGF8-based thresholding mechanism to determine cell fates. However, this model does not explain how cell determination and subsequent differentiation translates into somite morphology. In this paper, we use computer simulations of chick somitogenesis to show that experimentally-observed temporal and spatial patterns of adhesive N-CAM and N-cadherin and repulsive EphA4–ephrinB2 pairs suffice to reproduce the complex dynamic morphological changes of somitogenesis in wild-type and N-cadherin ($-/-$) chick, including intersomitic separation, boundary-shape evolution and sorting of misdifferentiated cells across compartment boundaries. Since different models of determination yield the same, experimentally-observed, distribution of adhesion and repulsion molecules, the patterning is independent of the details of this mechanism.

I. Introduction

Somitogenesis, during which initially-continuous anterior–posterior (AP) bands of loosely-bound cells on either side of the medial primitive streak, the *presomitic mesoderms* (PSM), break apart sequentially and periodically (at intervals of ~90 minutes in chick) in AP sequence into a spatially-regular series of separated, tightly-bound *somites* is the classical example of *segmentation* during vertebrate embryogenesis. In vertebrates, somites are the precursors of vertebrae, muscle and skin derivatives, and provide a scaffold for assembly of the peripheral vasculature and nervous system (Gossler and Hrabe de Angelis, 1998).

Somite formation requires: (i) Physical separation of somitic tissue from the initially-continuous PSM, (ii) coalescence of cells in the forming somite, and (iii) the establishment of a stable border between the somite and the PSM.

Experimentally, animal species differ in the cell rearrangements which create the somite–PSM border and how aggressively new somites pull apart from the PSM (Kulesa *et al.*, 2007). In *Xenopus* embryos, two short, discrete fissures start from both the medial and lateral edges of the PSM and expand gradually towards the middle of the PSM to form a stable somite–PSM border (Afonin *et al.*, 2006). In zebrafish embryos, cell rearrangements within forming somites are minimal and the forming-somite–PSM border develops when cells in the forming somite gently detach from their PSM neighbors, forming a medial notch which spreads laterally (Wood and Thorogood, 1994; Henry *et al.*, 2000; Jiang *et al.*, 2000). In both animals, the cells then retract towards the center of the forming somite or towards the PSM depending on the side of the somite–PSM boundary.

In chick embryos, cell rearrangements during somite–PSM border formation are more dramatic. Somitogenesis occurs in a complex spatiotemporal pattern, not via simple cleavage of the PSM (Kulesa and Fraser, 2002). Time-lapse analyses reveal that the forming somite–PSM boundary develops a dynamic *ball-and-socket* shape and that some cells cross the presumptive somite–PSM boundary (Kulesa and Fraser, 2002). Tissue transplantation studies have shown that cells in the region near the posterior border of the forming somite possess border-forming signals mediated by Notch and reinforced by Lunatic Fringe (Sato *et al.*, 2002). Transplantation of the ventral-most cells in the posterior of the forming somite induces formation of ectopic borders and somite subdivisions in more dorsal PSM tissues (Sato and Takahashi, 2005). Thus, in chick, the initial separation of a forming somite from the PSM appears to occur in a ventral-to-dorsal (VD) direction. However, in line with most experimental and computational studies, this paper will treat the PSM and somites as essentially two-dimensional (2D), neglecting VD and medial-to-lateral (ML) variation.

To explain the complex cell rearrangements in chick somitogenesis we employ computer simulations, which allow us to study how previously-determined spatiotemporal variations in gene expression (*prepatterns*) lead to variations in cell adhesion and local microenvironment, which induce cell rearrangement into coherent somites. The rearrangements, in turn, can further affect gene expression (*feedback*).

Essentially all models of somitogenesis to date have neglected the properties and movements of individual cells, and concentrated on mechanisms to generate periodic patterns of gene expression in the PSM, i.e., *somite specification*. This focus on early gene expression seemed reasonable, because determination of cell fate occurs early, about three somitic-clock cycles (about $4\frac{1}{2}$ hours) before physical segmentation (Dubrulle and Pourquié, 2004). The existence of a prepattern is consistent with the finding that reversal of the AP axis of the PSM leads to reversed somites (Keynes and Stern, 1988). The numerous theoretical models of gene-expression patterning include (Schnell and Maini, 2000; Baker *et al.*, 2003, 2006), the *clock-and-wavefront* model (Cooke and Zeeman, 1976; Dubrulle *et al.*, 2001), *reaction–diffusion* models (Meinhardt, 1996), *cell-cycle* models (Primmett *et al.*, 1988, 1989; Stern *et al.*, 1988; Collier *et al.*, 2000), and the *clock-and-induction* model (Schnell and Maini, 2000). Each of these models includes certain key features of the underlying biology and predicts that the PSM develops spatially-periodic patterns of gene expression in tissue blocks, but fails to explain some experimental observations. Pourquié and co-workers' version of the clock-and-wavefront model (Durbrulle and Pourquié, 2002; Pourquié, 2004) is, perhaps, the most successful (Baker *et al.*, 2006). For a detailed review of mathematical models of somitic prepatterning, see the paper by Baker *et al.* in this volume.

The actual process of somite formation—how a somite pulls apart from the PSM and the ensuing morphological changes—is not well understood. The only mathematical model attempting to describe the bulk movement of somitic cells to form a somite (Schnell *et al.*, 2002) does not account for the intercellular mechanical forces involved in somite formation. Grima and Schnell (2007) have investigated the possibility that minimization of tissue surface tension drives the subsequent morphological changes leading to rounded somites. This paper addresses another question—given a prepattern of gene expression, can known biological mechanisms give rise to the patterns of cell movement and morphological changes observed in both wild-type and gene-knockout experiments? Our model is completely agnostic about the origin of this pattern of gene expression (it works equally well with any of the somite-specification models in the preceding paragraph).

A. Nomenclature

Because somitogenesis proceeds in a temporally-periodic and spatially-progressive fashion, the identity of groups of cells changes in time, making nomenclature somewhat confusing (see Fig. 2). We use the nomenclature most common in the experimental community. At the beginning of a 90-minute cycle of somite formation, the anterior-most portion of the PSM becomes the site of a *newly-forming somite*, which we refer to as S_0; we will then refer to the portion of PSM immediately posterior to the region which contains the forming somite as the *anterior* PSM or S_{-1}. Initially, these two regions are contiguous; the forming somite S_0 then gradually separates from the remaining PSM along the *presumptive somite–PSM boundary*. When somite separation is complete, the forming somite S_0 becomes somite S_1, the anterior portion of the remaining PSM, S_{-1}, becomes S_0 and the cycle repeats (Ordahl, 1993; Pourquié and Tam, 2001). We call the regions of the PSM successive somite lengths behind S_{-1}, S_{-2}, S_{-3}, ... Each somite also has an *anterior* and *posterior compartment*, which we will denote S_{-1A} and S_{-1P}, respectively, and a *central* (or *core*) and *peripheral* region.

II. Patterns of Gene Expression and Protein Distribution during Somitogenesis

To understand how the gene-expression prepattern translates into changes in tissue morphology, we first review the functions of key molecules and their expression patterns.

A. Cell Adhesion Molecules

N-CAM and N-cadherin are *homophilic* membrane-bound proteins which contribute to contact adhesion between cells. The strength of adhesion increases with protein density on each cell, though the form of the dependence is not completely clear (Foty and Steinberg, 2005). While both molecules are homotypically adhesive as isolated monomers, they normally associate into groups which then aggregate in

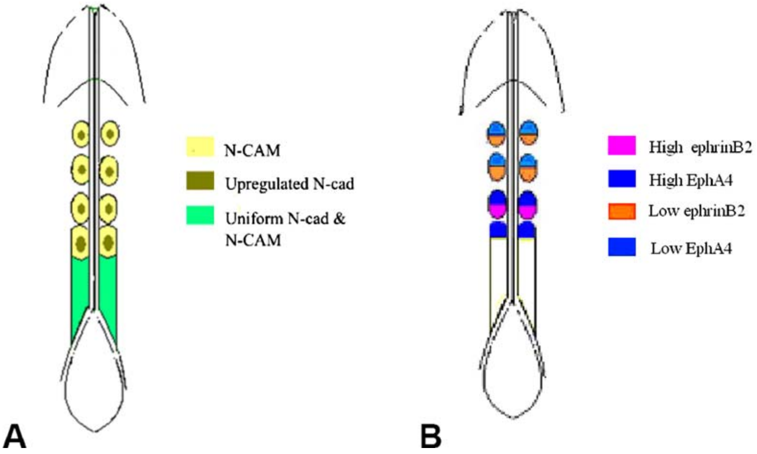

Figure 1 Schematic diagram of the distribution of adhesion and repulsion molecules during chick somitogenesis. (A) N-cadherin and N-CAM protein distributions based on immunocytochemistry experiments (Linask *et al.*, 1998). The PSM has a uniform low background level of N-cadherin and N-CAM. N-CAM levels do not change significantly during segmentation, while N-cadherin levels increase at the core of the forming somite. (B) EphA4 and ephrinB2 levels based on *in-situ*-hybridization experiments (Baker *et al.*, 2003; Aulehla and Pourquié, 2006). During segmentation EphA4 expresses in the anterior compartment of the forming somite and the anterior of the PSM; ephrinB2 expresses in the posterior compartment of the forming somite. Both EphA4 and ephrinB2 mRNA levels decrease after the somite has separated from the PSM. See color insert.

the cell membrane to form clusters (e.g., clusters of trimers of dimers) which increases the effective binding strength per molecule. Both molecules bind to the actin cytoskeleton via β-catenin. This binding affects their effective adhesivity directly, possibly through changes in conformation of their extracellular domains, and indirectly, because a functional actin cytoskeleton is necessary for their clustering. In general, for a given receptor density, N-CAM results in weaker adhesion than N-cadherin.

In avian and mouse embryos, somite formation follows *compaction* (i.e., a reduction in the intercellular space between cells) and the heightened expression of N-CAM and N-cadherin (Duband *et al.*, 1987; Linask *et al.*, 1998), which are expressed at lower levels in the rest of the PSM (Fig. 1A). In mice (Kimura *et al.*, 1995), cadherin-11 strictly correlates with S_0; it is not expressed in other parts of the PSM. Fig. 1A schematically illustrates the dynamic changes of cell-adhesion-molecule types and concentrations during somite segmentation in chick embryos. Before segmentation, mesenchymal cells in the PSM weakly express N-CAM and N-cadherin. During segmentation, the condensing cells in S_0 significantly increase their N-CAM and N-cadherin expression.

Before and during somite separation, the cells at the periphery of S_0 *epithelialize* (Dubrulle and Pourquié, 2004). During this epithelialization, N-CAM remains uniformly distributed over the entire surfaces of epithelial cells, whereas N-cadherins concentrate predominantly on the apical surfaces (which somewhat counter-intuitively face towards the center of the somite). Cells located in the core of the somite remain mesenchymal and continue to have essentially uniform surface distributions of both N-CAM and N-cadherin (Duband *et al.*, 1987; Linask *et al.*, 1998).

B. Eph/ephrin-Induced Cell "Repulsion"

Ephs and ephrins are families of heterotypically-active cell-surface receptors that can lead to effective "repulsion" between a cell expressing an Eph and a cell expressing the corresponding ephrin. How contact leads to effective repulsion is still under active investigation. A plausible mechanism is that pairing of complementary Ephs and ephrins on apposing cells triggers bidirectional signaling, which results in the local collapse of the actin cytoskeleton in both cells near their point of contact (Harbott and Nobes, 2005). This collapse then locally disrupts the condensation and pairing of cell adhesion molecules like N-CAM and N-cadherin, which reduces, but need not completely eliminate, the strength of their homotypic intercellular binding (as suggested in Kasemeier-Kulesa *et al.*, 2006; Cooke *et al.*, 2005). Thus EphA4–ephrinB2 "repulsion" in somites may actually result from a reduction in effective cell–cell adhesion. Since Eph–Eph and ephrin–ephrin apposition has no effect on the cytoskeleton, cell adhesion molecules remain fully functional within the interior of an Eph-expressing or ephrin-expressing domain. Thus boundaries between domains expressing an Eph and its complementary ephrin are structurally weak, while the domains themselves can be strong, allowing tissue to pull apart along Eph–ephrin contact lines. Collapse of the actin cytoskeleton also destroys the *pseudopods* or *leading edges* which cells use to move in a particular direction (Mellitzer *et al.*, 1999; Xu *et al.*, 1999; Poliakov *et al.*, 2004), preventing a cell expressing Eph from moving into a cluster of cells expressing the corresponding ephrin and *vice versa* (*contact inhibition*). The net effect is to establish compartmental boundaries between clusters of cells expressing Eph and clusters of cells expressing the complementary ephrin. Such boundaries are clearly visible using ordinary microscopy.

Eph/ephrin signaling is responsible for boundary formation in the developing hindbrain (Mellitzer *et al.*, 1999; Xu *et al.*, 2000) and is necessary for the formation of intersomitic boundaries and subsequent epithelialization (Durbin *et al.*, 1998). During somite segmentation, EphA4 expresses in the anterior half of somites (S_{0A}) and in the anterior tip of the PSM (S_{-1A}), while EphrinB2 expresses in the posterior half of somites (S_{0P}) (Fig. 1B) (Nieto *et al.*, 1992; Bergemann *et al.*, 1995; Durbin *et al.*, 1998; Baker and Antin, 2003; Baker *et al.*, 2003). EphA4/ephrinB2 signaling also regulates the mesenchymal-to-epithelial transition of the PSM during somitogenesis (Barrios *et al.*, 2003).

C. Interaction Between Adhesion and Repulsion during Somitogenesis

That the adhesive interactions of N-CAM and N-cadherin are homophilic while EphA4 and ephrinB2 produce an effective *heterorepulsion* (repulsion between cells of two complementary types) is crucial to the observed mechanics of somite formation. Cells expressing EphA4 and cells expressing ephrinB2 meet both at the presumptive somitic boundary (S_{0P} to S_{-1A}) and inside each somite (at the center of S_0, i.e., S_{0A} to S_{0P}). If contact between cells expressing EphA4 and cells expressing ephrinB2 causes the separation of the posterior end of S_0 (S_{0P}) from the anterior end of the PSM (S_{-1A}), why does it not cause a similar boundary to form inside S_0 (between S_{0A} and S_{0P}), subdividing it into two noncontacting smaller somites? N-cadherin in the center of the somite seems to be essential, since, in mouse, knocking out N-cadherin results in segmentation of normal somites into two separated sub-somites (Kimura *et al.*, 1995; Radice *et al.*, 1997; Horikawa *et al.*, 1999). One question we will investigate through simulation is what interactions of adhesion and repulsion lead to segmentation without fragmentation of individual somites.

III. From Genetic Oscillators to Adhesion/Repulsion-Protein Patterns

Molecular signaling during segmentation-prepattern specification is still a subject of intensive research. The following highly-simplified and speculative description provides at least a working hypothesis for these mechanisms. The embryo elongates primarily through division of cells in the extreme posterior of the PSM and in the tail bud. Cells leave the tail bud, then cease to transcribe fgf8 DNA into mRNA. This mRNA slowly degrades, but continues to be translated into FGF8 protein (Dubrulle and Pourquié, 2004); thus the average level of FGF8 in more anterior PSM cells is lower than in more posterior cells. In addition, the precursor to retinoic acid (RA), RALDH2, diffuses into the PSM from the anterior of the embryo. FGF8 and RA couple antagonistically within cells to select either high-RA or high-FGF8 states in a bistable manner (Diez del Corral *et al.*, 2003; Goldbeter *et al.*, 2007). Above a threshold level of FGF8, individual cells in the PSM exhibit spontaneous oscillations in expression levels of many genes, especially those belonging to the Notch, Wnt and FGF cascades (Palmeirim *et al.*, 1997; Dequeant *et al.*, 2006; Kulesa *et al.*, 2007), with a period which depends on cytoplasmic FGF8 levels. Neighboring cells' oscillators synchronize via Delta–Notch signaling (Horikawa *et al.*, 2006). The oscillation of FGF8 activity is superimposed on the background decrease of FGF8, a somewhat unusual circuit that results in a whole block of cells (the size of one somite, or about 200 microns in chick) switching to the high-RA state nearly simultaneously. The combination of this switch with the local phase of the oscillation determines the cell's later differentiation (Aulehla *et al.*, 2003; Aulehla and Herrmann, 2004). Oscillations cease some time after this switch, but their

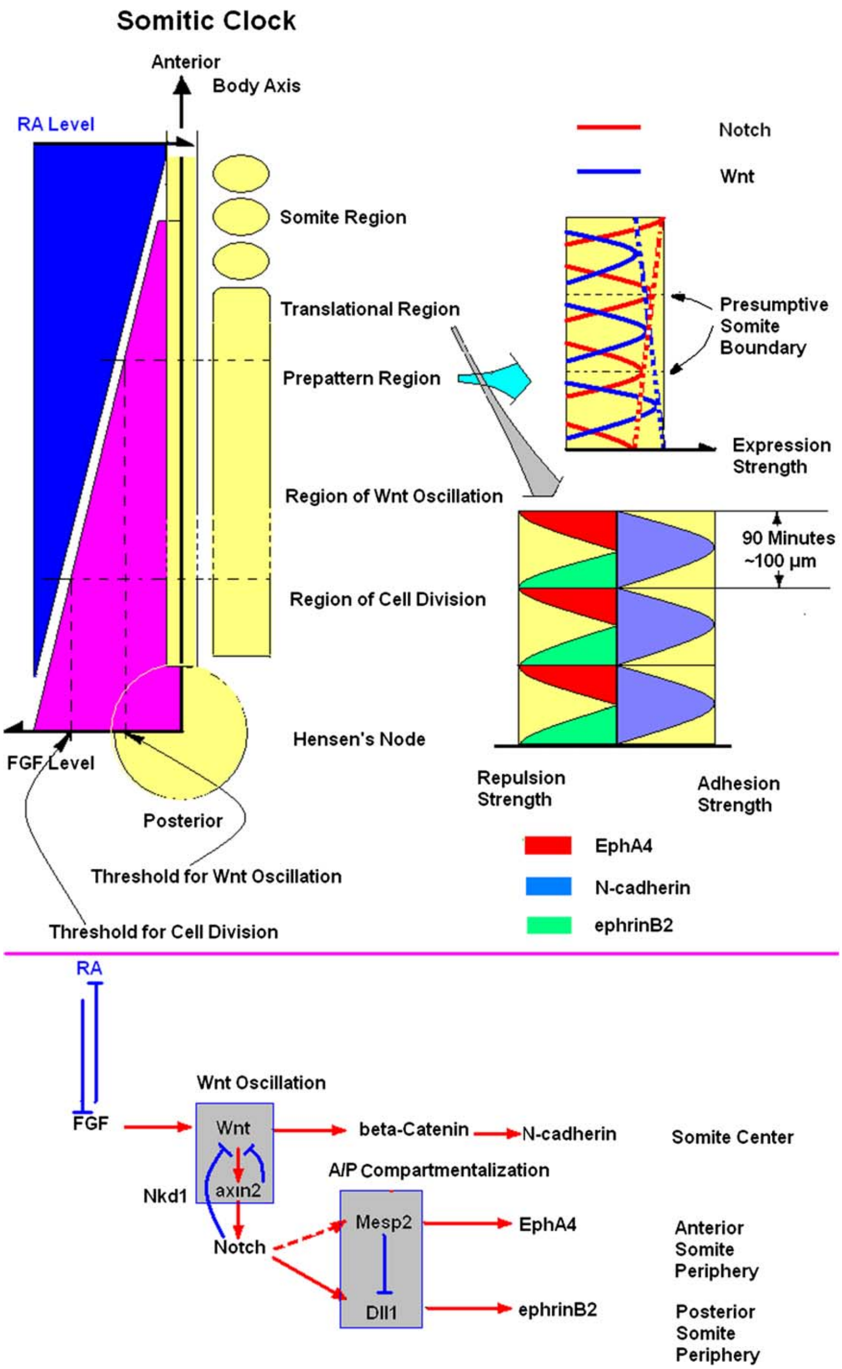
Somitic Clock
Anterior
Body Axis
RA Level
Notch
Wnt
Somite Region
Translational Region
Prepattern Region
Presumptive
Somite
Boundary
Expression
Strength
Region of Wnt Oscillation
90 Minutes
~100 µm
Region of Cell Division
Hensen's Node
FGF Level
Posterior
Threshold for Wnt Oscillation
Threshold for Cell Division
Repulsion
Strength
Adhesion
Strength
EphA4
N-cadherin
ephrinB2
RA
Wnt Oscillation
FGF
Wnt
beta-Catenin
N-cadherin
Somite Center
A/P Compartmentalization
Nkd1
axin2
Notch
Mesp2
EphA4
Anterior
Somite
Periphery
Dll1
ephrinB2
Posterior
Somite
Periphery

Figure 2 Schematic diagram of the translation of FGF8, Wnt and Notch cyclic expression into spatially-periodic patterns of N-cadherin, EphA4, and ephrinB2. (Top left) Within the PSM, fgf8-mRNA decay produces a PA decrease in FGF8 protein levels. Above a threshold FGF8 concentration, the intracellular Wnt concentration oscillates with a period of ~90 minutes. Intercellular Delta-Notch signaling (not shown) synchronizes neighboring cells, producing coherent oscillations in Wnt concentration. (Top right) When the FGF8 concentration falls below a threshold, Wnt oscillations cease, creating a spatial oscillation in Wnt levels. The inhibitory interaction between Wnt and Notch leads to out-of-phase expression of Wnt and Notch. (Middle right) N-cadherin, EphA4, and ephrinB2 levels. N-cadherin expression is maximal at the cores of somites. EphA4 is expressed in anterior somite compartments and ephrinB2 in posterior compartments. Both are maximal at somite boundaries. (Bottom) Plausible regulatory links from Wnt and Notch to N-cadherin, EphA4, and ephrinB2. The mutual inhibition of Wnt and Notch by axin2 and Nkd1 causes the Wnt and Notch oscillations. In the anterior somite, Mesp2 suppresses Notch signaling and also induces EphA4 expression. In the posterior somite, DLL1 maintains ephrinB2. See color insert.

persistence has no known effect on cell differentiation (Pourquié, 2007, personal communication).

Phase read-out during cell determination seems to depend on Wnt signaling (Aulehla and Pourquié, 2006), though the full regulatory cascades are not known. Fig. 2 summarizes regulatory interactions (bottom) and AP expression patterns (top). The fundamental prepatterning mechanism is the inhibitory coupling from Notch to Wnt, via axin2, Nkd1, and other pathways, which leads to elevated levels of Wnt in the centers of presumptive somites and of Notch at presumptive somite boundaries (Pourquié, 2007, personal communication). The expression domains are fairly broad, with substantial overlap in the regions midway between a presumptive somite's center and its boundaries. Wnt stabilizes cytoplasmic β-catenin, which acts as a transcription factor increasing expression of N-cadherin, which accumulates in a broad area around the presumptive somite's core (Nelson and Nusse, 2004). Notch similarly transiently upregulates Eph-ephrin expression and activity at the presumptive somite boundaries. The selective expression of the transcription factor Mesp2, activated by RA (Moreno and Kintner, 2004), in the anterior half of each presumptive somite, defines anterior and posterior compartments by inhibiting DLL expression in the anterior compartments (Takahashi *et al.*, 2003). Mesp2 further leads to transient expression of EphA4 in the anterior compartments (Nakajima *et al.*, 2006), which peaks near the presumptive anterior boundary. In the posterior compartments, DLL maintains ephrinB2 expression (De Bellard *et al.*, 2002), peaking near the presumptive boundary. Other protein level changes result from unknown mechanisms. The time lag from determination to segmentation seems primarily to result from transcriptional and translational delays and the time required for EphA4, ephrinB2, N-CAM and N-cadherin to reach functional levels (Kulesa, 2007, private communication). EphrinB2 levels seems to increase more slowly than do the levels of the other molecules. We have included this difference in the timing of molecular level changes in our simulations; however, the exact sequence is not crucial to our results.

IV. From Adhesion-Protein Patterns to Segmentation

Based on experiments that show that cells in the PSM condense into somites by changing their adhesive and migratory properties (Gossler and Hrabe de Angelis, 1998), our key hypothesis is that the primary ways that the cells' internal differentiation states translate into mechanical activity are through:

i) Differential expression and binding of the cell adhesion molecules N-CAM and N-cadherin.
ii) Differential expression and bidirectional activation of EphA4 receptors and ephrinB2 proteins.

Segmentation then results from the spatiotemporal coordination of N-CAM, N-cadherin and Eph-ephrin expression.

A. Segmentation Model

We use a 2D approximation, neglecting DV variations. We also neglect mediolateral (ML) variations in cell properties (which may be quite important in many cases), assume that the intrinsic level of cytoskeletal cell motility is constant in all cells and that cells do not divide or die.

Based on the biological observations we described above, our model assumes that somitogenesis depends predominantly on four molecular species: N-CAM, N-cadherin, EphA4 and ephrinB2. We make a number of simplifying assumptions concerning the behavior and spatiotemporal expression patterns of these molecules. We assume that the distribution of these adhesive and repulsive species is uniform over cell membranes and that the primary effect of peripheral epithelialization is to change the relative cell sizes and adhesions in the somite core and periphery rather than redistributing adhesion molecules over cell surfaces. We assume that binding of EphA4 and ephrinB2 on apposing cells reduces the effective adhesion due to their cell adhesion molecules. However, we do include the variation in levels of molecular expression between the core and periphery of each somite and between the anterior and posterior halves of each somite compartment.

We assume the following temporal sequence of molecular distributions (see Fig. 3). All molecular levels turn on abruptly at the beginning of a segmentation cycle (times $t_0, t_1, t_2, t_3, \ldots$) and remain the same until the end of the simulation. In reality, these levels would change further as the somites matured, but the chief focus of this paper is the initial formation of the somites rather than their later maturation. We do not attempt to model the origin of the somite-cycle timing. Within the posterior PSM (S_{-2}, S_{-3}, ...), all cells express a uniform background level of N-CAM and N-cadherin, keeping the cells loosely connected. At the beginning of each segmentation cycle, N-cadherin levels in the cells at the core of S_{-1} (the somite posterior to the forming somite) increase substantially, N-cadherin levels in cells at the periphery of S_{-1} decrease, high levels of EphA4 appear in the anterior half of the anterior compartment ($\mathrm{S}_{-1\mathrm{A}}$)

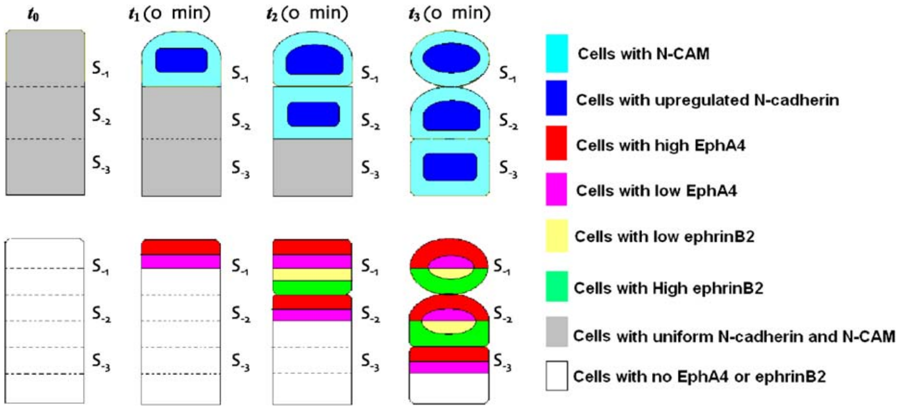

Figure 3 Schematic diagram of the spatiotemporal activation of N-cadherin, N-CAM, EphA4, and ephrinB2 during somite segmentation. (Upper panels) At t_0, all cells have uniform N-cadherin and N-CAM levels. At t_1, N-cadherin levels increase in the core of the anterior domain of the PSM (the new somite S_{-1}). At t_2, when the somitic boundary between S_0 and S_{-1} starts to form, N-cadherin levels increase in the core of somite S_{-1}. (Lower panels) At t_0, no cells have EphA4 or ephrinB2. At t_1 the cells in the anterior compartment of S_{-1} increase their EphA4 level, with a high level in the anterior half of the compartment and a low level in the posterior half. At t_2 the cells in the anterior compartment of the new S_{-1} increase their EphA4 level, with a high level in the anterior half of the compartment and a low level in the posterior half, while the cells in the posterior compartment of S_0 increase their levels of ephrinB2, with a high level in the posterior half of the compartment and a low level in the anterior half. (Both panels) At t_3, an intersomitic boundary forms between S_0 and S_{-1}, and the expression of N-cadherin, EphA4, and ephrinB2 reiterates caudally. See color insert.

and low levels in the posterior half of the anterior compartment (S_{-1A}). To represent the delayed appearance of ephrinB2, we also turn on a low level of ephrinB2 in the anterior half of S_{0P}, and a higher level in the anterior half of S_{0P}.

We assume that cells in the PSM initially have the same size. We represent the epithelialization and compaction of the peripheral cells into *epithelial cells*, by increasing the size of peripheral cells and decreasing the size of core cells slightly when a given compartment begins to express either EphA4 or ephrinB2.

At the following cycle (t_2, t_3, . . .), this entire pattern repeats, shifted posteriorly by one somite length.

V. Computer Simulation of Segmentation

A. The Glazier–Graner–Hogeweg Model

To simulate our somite-segmentation model, we implement it using the *Glazier–Graner–Hogeweg* model (GGH) (also known as the *Extended* or *Cellular Potts Model*) (Glazier and Graner, 1992, 1993). Extensive comparisons between experiments and

GGH simulations have validated GGH methods (Mareé and Hogeweg, 2001; Zeng *et al.*, 2004; Chaturvedi *et al.*, 2005; Poplawski *et al.*, 2007; Merks *et al.*, 2006; Merks and Glazier, 2006) for multicell morphogenesis modeling.

Cells in the GGH are extended domains of pixels on a lattice which share the same *cell index* σ. They can represent either biological cells or *generalized cells* like subregions of extracellular matrix (ECM). We define an *Effective Energy*, which includes the key biological features we wish to simulate (in this case, cell adhesion and cells' volumes and membrane areas):

$$\begin{aligned} H = & \sum_{\substack{\vec{i},\vec{i}' \\ \text{neighbors}}} J\big(\tau\big(\sigma(\vec{i})\big), \tau\big(\sigma(\vec{i}')\big)\big)\big(1 - \delta\big(\sigma(\vec{i}), \sigma(\vec{i}')\big)\big) \\ & + \sum_{\sigma} \lambda\big(V\big(\sigma(\vec{i})\big) - V_t\big(\tau\big(\sigma(\vec{i})\big)\big)\big) \\ & + \sum_{\sigma} \beta\big(S\big(\sigma(\vec{i})\big) - S_t\big(\tau\big(\sigma(\vec{i})\big)\big)\big), \end{aligned} \tag{1}$$

where J is the total adhesion energy per unit surface area between cells of type $\tau(\sigma)$ and $\tau(\sigma')$ (negative for adhesion), V_t the target volume of cells of type τ, V the actual volume of each cell, λ the strength of the volume constraint (the bigger λ the smaller the cells' volume fluctuations), S_t the target membrane area for cells of type τ, S the actual membrane area of each cell, and β the membrane elasticity.

We also define a dynamics which represents cell motility. We pick a lattice site, $\vec{i}$, at random and have one of the cells occupying a neighboring lattice site, $\vec{i}'$, attempt to displace the cell currently occupying that site. If the displacement lowers H, we accept it. Otherwise, we accept the displacement with a probability decaying exponentially in the Effective Energy cost, rescaled by the *Effective Cell Motility* $T_m > 0$:

$$P\big(\sigma(\vec{i}) \to \sigma(\vec{i}')\big) = \big\{\exp(-\Delta H/T_m\colon\ \Delta H > 0;\ 1\colon\ \Delta H \leqslant 0)\big\}. \tag{2}$$

If our lattice has N pixels, we define one *Monte-Carlo Step* (MCS) to be N displacement attempts. We discuss the conversion between MCSs and experimental time below.

B. GGH Somitogenesis Simulation

To translate our model into a GGH simulation, we must define the GGH parameters and their temporal changes for each cell.

Fig. 4 shows the initial condition of our 2D rectangular lattice (170 × 450 pixels). We begin with a segment of PSM, with ECM surrounding the PSM in the anterior and lateral directions. The segment of PSM, which extends to the bottom of the lattice, is long enough to allow more than three somites to form and the ECM layer is thick enough that the cells do not interact with the lattice edges in the region of somite

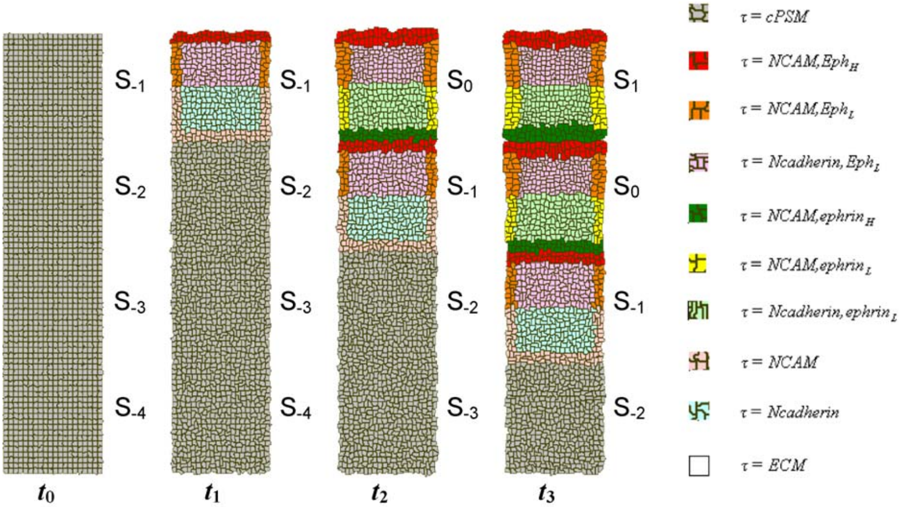

Figure 4 Simulated spatiotemporal levels of N-cadherin, N-CAM, EphA4, and ephrinB2 during somitogenesis. At $t_0 = 0$ MCS, the simulation begins with a regular array of PSM cells expressing background levels of N-CAM and N-cadherin, and surrounded by ECM. At $t_1 = 5000$ MCS, the N-cadherin level increases in the core of S_{-1} and the N-CAM level increases at the periphery. EphA4 levels increase to High in the anterior half of S_{-1A} and to Low in the posterior half, while the peripheral cells in S_{-1A} grow slightly in volume and the core cells shrink slightly in volume. At $t_2 = 7000$ MCS this process repeats in the new S_{-1} and the ephrinB2 level increases to High in the posterior half of S_{-0P} and to Low in the anterior half, while the peripheral cells in S_{-0P} grow slightly in volume and the core cells shrink slightly in volume. At $t_3 = 9000$ MCS, the process repeats for the new S_0 and S_{-1} somites. See color insert.

formation. The long axis of the rectangle corresponds to the AP axis in the embryo and the short axis to the ML axis. Each cell initially occupies 5 × 5 pixels and each somite contains 20 × 20 cells, which corresponds to the approximately 400 cells in chick somites (Kulesa and Fraser, 2002).

Fig. 4 shows our 10 cell types:

1. ECM substrate (a single large generalized cell with unconstrained volume), $\tau = ECM$.
2. Mesenchymal cells in the posterior PSM with uniform volumes and low background levels of N-cadherin and N-CAM, $\tau = cPSM$.
3. Epithelial cells in the periphery of the anterior half of the anterior compartments of somites, with increased volumes, high levels of EphA4, background levels of N-CAM and low levels of N-cadherin, $\tau = NCAM, Eph_H$.
4. Epithelial cells in the periphery of the posterior half of the anterior compartments of somites, with increased volumes, low levels of EphA4, background levels of N-CAM and low levels of N-cadherin, $\tau = NCAM, Eph_L$.

5. Mesenchymal cells in the core of the posterior half of the anterior compartments of somites, with decreased volumes, low levels of EphA4, background levels of N-CAM and high levels of N-cadherin, $\tau = Ncadherin, Eph_L$.
6. Epithelial cells in the periphery of the posterior half of the posterior compartments of somites, with increased volumes, high levels of ephrinB2, background levels of N-CAM and low levels of N-cadherin, $\tau = NCAM, ephrin_H$.
7. Epithelial cells in the periphery of the anterior half of the posterior compartments of somites, with increased volumes, low levels of ephrinB2, background levels of N-CAM and low levels of N-cadherin, $\tau = NCAM, ephrin_L$.
8. Mesenchymal cells in the core of the anterior half of the posterior compartments of somites, with decreased volumes, low levels of ephrinB2, background levels of N-CAM and high levels of N-cadherin, $\tau = Ncadherin, ephrin_L$.
9. Mesenchymal cells in the anterior region of the PSM corresponding to the core of somite S_{-1}, with their original volumes, background levels of N-CAM and high levels of N-cadherin, $\tau = Ncadherin$.
10. Mesenchymal cells in the anterior region of the PSM which will correspond to the periphery of somite S_{-1}, with their original volumes, background levels of N-CAM and low levels of N-cadherin, $\tau = NCAM$.

A clock controls the temporal distribution of cell types. All cells except the ECM begin as type *PSM*. At the beginning of the first somite cycle in the simulation, at t_1, cells in the anterior region of the PSM, which we define as beginning as S_{-1A}, change their types from type *cPSM* to type $NCAM, Eph_H$, $NCAM, Eph_L$ or $Ncadherin, Eph_L$, and those in S_{-1P}, change their types from type *cPSM* to type *Ncadherin* or *NCAM*. At the beginning of each succeeding somite cycle, the cells in the new S_{0P} change their types from types *Ncadherin* or *NCAM* to types *NCAM*, *ephrinH*, *NCAM*, *ephrinL* or *Ncadherin*, *ephrinL*, cells in S_{-1A} change from type *cPSM* to type $NCAM, Eph_H$, $NCAM, Eph_L$ or $Ncadherin, Eph_L$, and cells in S_{-1P}, change from type *cPSM* to type *Ncadherin* or *NCAM*. Fig. 4 shows the spatiotemporal activation of NCAM, N-cadherin, Eph, and ephrin during the simulation, which schematically reproduces the biological pattern in Fig. 3.

C. Simulation Implementation

We implemented our simulations using the open-source software package CompuCell3D, downloadable from (https://simtk.org/home/compucell3d and http://compucell3d.org/), which allows rapid translation of complex biological models into simulations using a combination of CC3D ML and Python scripting. The great advantage of this framework is that it allows compact description of models and hence their publication and validation. We provide our simulation code in Appendices A, B, and C.

Table I Initial and Target Values for Surface Areas and Volumes of Specific Cell Types

τ	λ	V	Biological Value	V_t	Biological Value	β	S	S_t
ECM	0	NA	NA	NA	NA	0	NA	NA
NCAM	20	25	100 μm^2	25	100 μm^2	20	20	20
Ncadherin	20	25	100 μm^2	25	100 μm^2	20	20	20
$NCAM, Eph_H$	20	25	100 μm^2	36	144 μm^2	20	20	24
$NCAM, Eph_L$	20	25	100 μm^2	36	144 μm^2	20	20	24
$Ncadherin, Eph_L$	20	25	100 μm^2	16	64 μm^2	20	20	16
$NCAM, ephrin_H$	20	25	100 μm^2	36	144 μm^2	20	20	24
$Ncadherin, ephrin_L$	20	25	100 μm^2	16	64 μm^2	20	20	16
$NCAM, ephrin_L$	20	25	100 μm^2	36	144 μm^2	20	20	24
cPSM	20	25	100 μm^2	25	100 μm^2	20	20	20

Table II Global Simulation Parameters

	Simulation Value	Biological Value
T_m	100	
t_1	5000 MCS	0 minutes
t_2	7000 MCS	90 minutes
t_3	9000 MCS	180 minutes
t_{total}	15,000 MCS	270 minutes
δ	7.0	14 μm

D. Parameter Values

Our GGH simulation has a substantial number of parameters, most of which are not known quantitatively from experiments. Fortunately, the patterning depends primarily on the hierarchy of the adhesion energies. Since our primary goals in this paper are qualitative reproduction of experiments, we retain a good deal of flexibility in choosing parameter values.

E. Global Parameters

The parameter T_m determines the intrinsic cell motility and rescales all of the other parameters in the Effective Energy. We are therefore free to fix it and vary the scale of our other parameters or *vice versa*. In this case we fix $T_m = 100$ unless we specify otherwise. We then choose the remaining parameters so that $\Delta H/T_m$ in Eq. (2) is neither too large nor too small (typically between about 0.5 and 5). If $\Delta H/T_m$ is very large, the cells will not move at all. If $\Delta H/T_m$ is large, cells will interact with the lattice, producing unnatural shapes, while, if $\Delta H/T_m$ is too small, cells may fall apart. In

Table III Initial Adhesion Energies (J) for Mesenchymal and Epithelial Cell Sorting during Somitogenesis (m indicates a mesenchymal cell type, e an epithelial cell type)

τ	τ									
	ECM	*NCAM*	*Ncadherin*	$NCAM, Eph_H$	$NCAM, Eph_L$	$Ncadherin, Eph_L$	$NCAM, ephrin_H$	$Ncadherin, ephrin_L$	$NCAM, ephrin_L$	*cPSM*
		m	*m*	*e*	*e*	*e*	*e*	*e*	*e*	*m*
ECM	0	15	15	15	15	15	15	15	15	15
NCAM		−20.25	−24	−20.25	−20.25	−24	−20.25	−24	−20.25	−20.25
Ncadherin			−38.44	−24	−24	−38.44	−24	−38.44	−24	−38.44
$NCAM, Eph_H$				−20.25	−20.25	−24	−20.25	−24	−20.25	−20.25
$NCAM, Eph_L$					−20.25	−24	−20.25	−24	−20.25	−20.25
$Ncadherin, Eph_L$						−38.44	−24	−38.44	−20.25	−38.44
$NCAM, ephrin_H$							−20.25	−24	−20.25	−20.25
$Ncadherin, ephrin_L$								−38.44	−24	−38.44
$NCAM, ephrin_L$									−20.25	−20.25
cPSM										−20.25

principle, we can calculate T_m from experiments measuring the diffusion constants of cells in the tissue, but these measurements are not yet available for somites. We chose the values of $\beta = 20$ and $\lambda = 20$ for the cell-volume and surface-area constraints to keep surface and volume fluctuations relatively small, without causing cells to stop moving. We present the parameters in Tables I, II, III.

F. Cell-Adhesion Energies

We have two types of adhesion in the simulation: cell–ECM adhesion and cell–cell adhesion.

G. Adhesion Hierarchies

The biology of the somite largely determines our adhesion hierarchy (where we have no information at all, we will set parameters equal to a default value).

In general, we will assume that adhesion between cells with higher levels of N-CAM or N-cadherin is stronger and that both N-CAM and N-cadherin are homotypically adhesive and that N-cadherin is more cohesive than N-CAM:

For our two classes of *epithelial* (e) and *mesenchymal* (m) cells surrounded by ECM, we need an initially-random mixture of the two cell types to sort stably, with the epithelial cells at the surface and mesenchymal cells condensing in the core, which requires the following relations among the adhesion energies (Glazier and Graner, 1993):

$$J(m,m) < \big(J(m,m) + J(e,e)\big)/2 < J(m,e) < J(e,e). \tag{3}$$

To prevent cells disassociating into the ECM, rather than sticking together, requires that (Glazier and Graner, 1993):

$$J(e,e) < J(e, ECM) \leqslant J(m, ECM). \tag{4}$$

To prevent cells from dispersing into the ECM, we set $J(\mathit{cell}, \mathit{ECM}) = 15$ initially.

The interaction energy per unit contact length between cells includes both adhesion and *effective repulsion*. Binding between EphA4 and ephrinB2 on apposing cells reduces the effective adhesion (increases J) by an *effective repulsion energy*, J_r, compared to the adhesion energy, J_a, which we would predict based on the number and type of cell-adhesion molecules. The decrease in adhesion need not scale linearly with the number of bound Eph/ephrin pairs. The simplest approximation to such an effect is to write the *net adhesion*, J, as a sum of the adhesion and repulsion alone plus a bilinear perturbation of strength c:

$$J(\tau_1, \tau_2) = J_a(\tau_1, \tau_2) + J_r(\tau_1, \tau_2) + cJ_a(\tau_1, \tau_2)J_r(\tau_1, \tau_2). \tag{5}$$

When c is positive, the interaction weakens adhesion compared to the additive case. When c is negative, the interaction strengthens adhesion compared to the additive case.

Additionally, cells expressing EphA4 should not mix with cells expressing ephrinB2. In this case, the energy for cell–cell interaction with repulsion must be larger than that for cell–cell interaction with adhesion:

$$\begin{aligned} &J(Eph, cadherin, ephrin, cadherin)\\ &\quad > \big(J(Eph, cadherin, ECM) + J(ephrin, cadherin, ECM)\big)/2\\ &\quad = \big(J(cadherin, ECM) + J(cadherin, ECM)\big)/2. \end{aligned} \tag{6}$$

Since domains of EphA4 and ephrinB2 cells separate but remain compact as the ECM furrow forms between them, the ECM must engulf both domains:

$$\begin{aligned} &J(Eph, cadherin, Eph, cadherin) < J(Eph, caderin, ECM),\\ &J(ephrin, cadherin, ephrin, cadherin) < J(ephrin, cadherin, ECM). \end{aligned} \tag{7}$$

H. Target Areas and Volumes

In 2D, the ratio of a cell's target membrane area squared to its target volume determines how "floppy" the cell is. If the ratio is small, cells will be round and stiff; in the opposite limit, cells will be floppy and extended, like an uninflated beach ball.

I. Time

The conversion between MCS and experimental time will depend on the average value of $\Delta H/T_m$, and hence on the choices of parameters. We make this assignment empirically, by observing the time in MCS that somite reorganization takes after we switch on the pattern of N-CAM, N-cadherin, EphA4, and ephrinB2 and set it to the corresponding experimental time. For the values of parameters we have chosen, 2000 MCS = 90 minutes (or 1 MCS = 2.7 seconds).

Because our initial configuration uses nonbiological, rectangular cells, we set $t_0 =$ 5000 MCS to allow the PSM to relax before we turn on the initial pattern of N-CAM and N-cadherin. Since the somitic clock interval is 2000 MCS, $t_1 = 7000$ MCS, $t_2 = 9000$ MCS, and so on.

VI. Results and Discussion

A. Parameter Choices

All viable developmental mechanisms must be relatively insensitive to fluctuations in key parameters. Biochemical redundancy is one approach to such robustness. Our

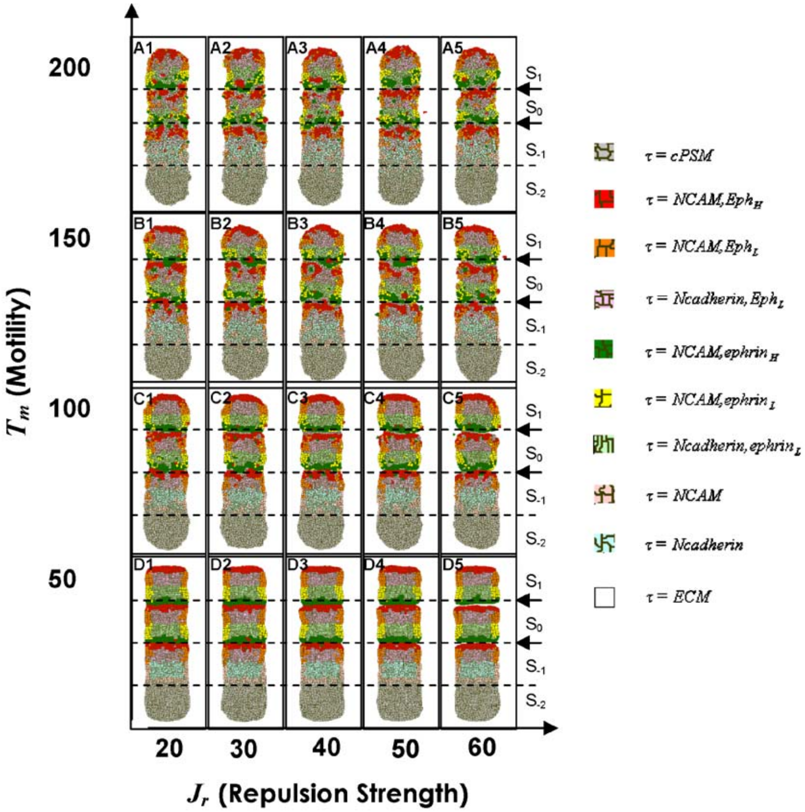

Figure 5 Morphology of somites for different cell motilities and EphA4/ephrinB2 repulsion strengths. Arrows (➡) mark intersomitic boundaries. (A1–A5), (B1–B5). For very high motilities, coherent somites do not form. (C1–C3), (D1–D2). Weak repulsion does not produce sharp intersomitic boundaries. (C5, D5). Segmentation with somite separation requires strong repulsion and limited cell motility. All simulations are shown after 15,000 MCS. See color insert.

model lacks such redundancy. However, even reasonable agreement with experiment would be unsatisfactory if it required very tight parameter tuning. Instead, we expect that a viable model will have a relatively broad range over which parameters have relatively little effect on segmentation. We therefore conducted parameter sweeps to explore parameter dependencies and optimize our choices.

Fig. 5 shows the long-time morphologies ($t = 15{,}000$ MCS) for cell motility T_m between 50 to 200 and repulsion energy J_r between 20 to 60. For very high cell motility $T_m \geqslant 150$ (Fig. 5A1–A5), coherent somites fail to form. Similarly, if the effect of EphA4/ephrinB2 binding is too weak ($J_r < 50$), the simulation does not form a clear intersomitic boundary. Only for strong repulsion ($J_r = 60$) and low motility ($T_m \leqslant 100$) does the somitic furrow form (Figs. 5C5 and 5D5). Thus, the sensitiv-

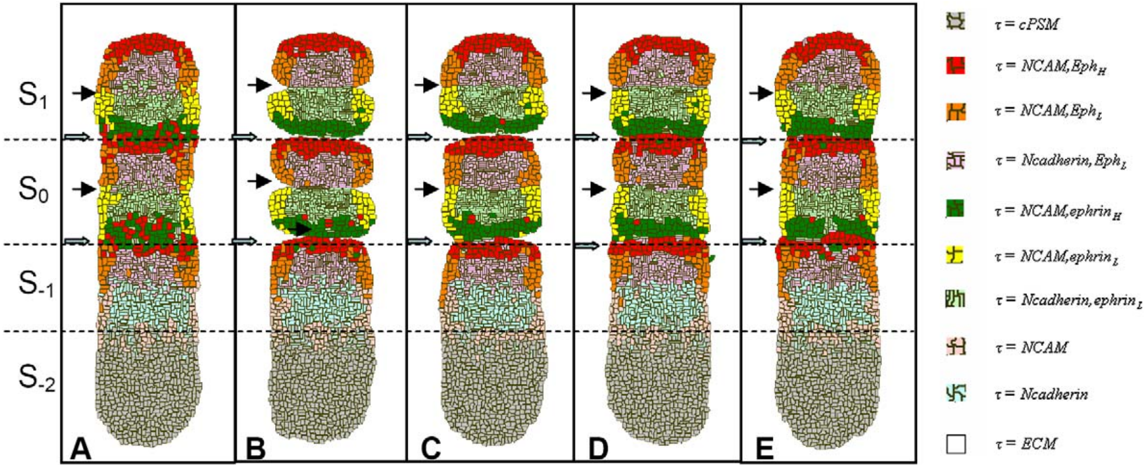

Figure 6 Effects of adhesion/repulsion coordination on somite morphology. Heavy arrows (⟹) mark intersomitic boundaries. Light arrows (➔) mark intrasomitic compartment boundaries. (A) Without EphA4 and ephrinB2, cells mix between the posterior of a somite and the anterior of the following somite. (B) Linear interaction of adhesion and repulsion with uniform repulsion results in both intersomitic and intrasomitic separation. (C) Linear interaction of adhesion and repulsion with graded repulsion results in intersomitic separation without intrasomitic separation. (D) Weak nonlinear interaction of adhesion and repulsion with uniform repulsion results in both intersomitic and intrasomitic segmentation. (E) Nonlinear interaction of adhesion and repulsion with graded repulsion results in intersomitic segmentation without intrasomitic segmentation. All simulations are shown after 15,000 MCS. See color insert.

ity to motility is fairly low (we can vary T_m by a factor of 2), but the sensitivity to variations in repulsion strength requires further study.

B. Segmentation Requires EphA4/ephrinB2 Repulsion

Fig. 6A shows the long-time configuration ($t = 15{,}000$ MCS) of a simulation in the absence of EphA4 and ephrinB2. The epithelial cells from different somites and mesenchymal cells from the anterior and posterior domains of each somite mix. Segmentation fails, as is observed in experiments in zebrafish which lack EphA4 (Barrios *et al.*, 2003).

C. Segmentation Requires Multiple Levels of EphA4/ephrinB2 Expression

We now explore the effect of EphA4/ephrinB2 repulsion on segmentation.

Our model includes two nonzero levels of EphA4 and ephrinB2: *High* and *Low*. We define J_{rHH} to be the repulsion energy between a cell with a High level of EphA4 and a cell with a High level of ephrinB2, J_{rHL} to be the repulsion energy between a cell with a High level of EphA4 and a cell with a Low level of ephrinB2 or between a cell with a Low level of EphA4 and a cell with a High level of ephrinB2, and J_{rLL} to be the repulsion energy between a cell with a Low level of EphA4 and a cell with a Low level of ephrinB2.

First, we test whether correct segmentation requires two levels of EphA4 and ephrinB2 expression within each somite compartment. We assume an additive relation between adhesion and repulsion ($c = 0$ in Eq. (5)) and that levels of EphA4 and ephrinB2 are uniform, $J_{rHH} = J_{rLL} = J_{rHL} = 60$. Fig. 6B shows that an intersomitic furrow (⇒) forms correctly. However an obvious intrasomitic notch (➔) develops between the anterior and posterior somite compartments, which does not occur in normal somite segmentation. Therefore, correct segmentation appears to require multiple levels of EphA4 and ephrinB2.

Next, we assume that the effect of the higher level of expression of EphA4 and ephrinB2 in the somite and PSM periphery results in effective repulsion energies of $J_{rHH} = J_{rHL} = 3J_{rLL} = 60$, still assuming an additive relation between adhesion and repulsion ($c = 0$ in Eq. (5)). Fig. 6C shows that an intersomitic furrow (⇒) forms correctly. A very small intrasomitic notch (➔) develops between the anterior and posterior somite compartments, as is observed experimentally. The resulting morphology is very close to normal somite segmentation.

Now we examine the effect of the bilinear term on the final pattern morphology. We assume a cooperative relation between adhesion and repulsion ($c = 1/50$ in Eq. (5)) and assume that High and Low levels of EphA4 and ephrinB2 are equal: $J_{rHH} = J_{rLL} = J_{rHL} = 60$. Even this small perturbation greatly changes the morphology from Fig. 6B. Fig. 6D shows that an intersomitic notch forms (⇒) but the somites do not separate. Somewhat surprisingly, the size of the intrasomitic notch (➔) is much smaller. Again, correct somite formation seems to require multiple levels of EphA4 and ephrinB2.

Finally, we assume a cooperative relation between adhesion and repulsion ($c = 1/50$ in Eq. (5)) and that the higher level of expression of EphA4 and ephrinB2 in the somite and PSM periphery results in effective repulsion energies of $J_{rHH} = J_{rHL} = 3J_{rLL} = 60$, as in Fig. 6C. Fig. 6E shows only a slight intersomitic notch (⇒) and no intrasomitic notch (➔). As in Figs. 6A and 6D, the somites do not separate.

Thus we will use additive repulsion ($c = 0$) at two levels, with $J_{rHH} = J_{rHL} = 3J_{rLL} = 60$ for our remaining simulations if we do not specify otherwise.

The Takeichi group (Horikawa *et al.*, 1999) has observed the separation of the anterior and posterior somite compartments during somite formation in N-cadherin $(-/-)$ mouse embryos. We simulated this experiment by setting the adhesion energy for cell types expressing N-cadherin equal to the adhesion energy for the corresponding cell types expressing N-CAM, effectively removing N-cadherin from our simulation. We kept the repulsion at the same levels as the values used for Fig. 6B. We observed both intersomitic (⇒) and intrasomitic (➔) furrows as seen in Fig. 7D. Thus N-cadherins seem essential to keeping the two compartments of a somite fused during segmentation.

D. Dynamic Morphological Changes and Error Correction during Segmentation

Time-lapse movies of shape changes during segmentation show intriguing effects (Kulesa and Fraser, 2002) (Figs. 8A–8F). In particular, the boundary between S_0 and S_{-1} forms a characteristic *ball-and-socket* or *W* shape (Fig. 8C), with the groove between S_0 and S_{-1} first opening up at the dips of the W, followed by the arms of the W retracting and sometimes folding inwards. Initially, the PSM envelops the forming somite, forming a *sleeve*. During segmentation, the sleeve cells fold back into the PSM along the S_0–S_{-1} boundary and the somite eventually rounds up.

One way to reproduce this dynamics would be for the PSM to cohere more to ECM than other cell types, i.e., $J(cPSM, ECM) < J(\ldots, ECM)$, in which case we expect that PSM cells will partially envelop the S_0 somite. A possible mechanism for such behavior would be a reduction in membrane levels of integrins in the posterior compartments of S_{-1} and later (S_0, S_1, ...) somites.

In Figs. 8G–8L, black arrows (➔) mark intersomitic boundaries (corresponding to the white arrows in Figs. 8A–8B). During the clock cycle before segmentation proper, and thus before the initiation of ephrinB2 expression in S_{-1} (Figs. 8G–8H), as in the experiment, we observe that anterior cells at the sides of S_{-2} move anteriorly and wrap around the posterior end of S_{-1}. During the following clock cycle (Figs. 8I–8J), the appearance of ephrinB2 in S_{0P} (formerly S_{-1P}) causes the formation of a W-shaped intersomitic furrow between S_0 and S_{-1}, as in Figs. 8C–8D. During the following cycle, the wings of the W retract and the somite rounds up (Figs. 8K–8L and 8E–8F).

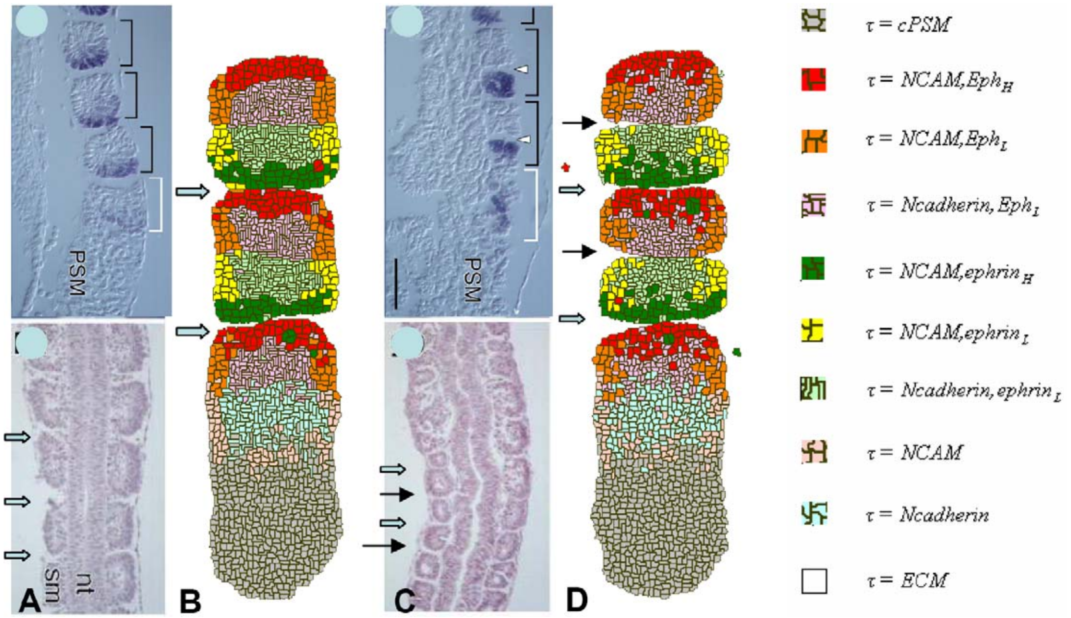

Figure 7 Comparison of somite structures in wild-type and N-cadherin-knockout experiments and simulations. Heavy arrows (⇒) mark intersomitic boundaries. Light arrows (➔) mark intrasomitic compartment boundaries. (A) Experimental wild phenotype (reprinted from Horikawa *et al.*, 1999, with permission from Elsevier). (B) Simulated wild phenotype after 15,000 MCS. (C) Experimental N-cadherin-double-knockout phenotype (reprinted from Horikawa *et al.*, 1999, with permission from Elsevier). The somites separate into *Uncx4.1*-positive and *Uncx4.1*-negative regions. *Uncx4.1* is a specific marker for the posterior compartments of somites. (D) Simulated N-cadherin-double-knockout phenotype at 10,000 MCS. Both somites and somite compartments separate. See color insert.

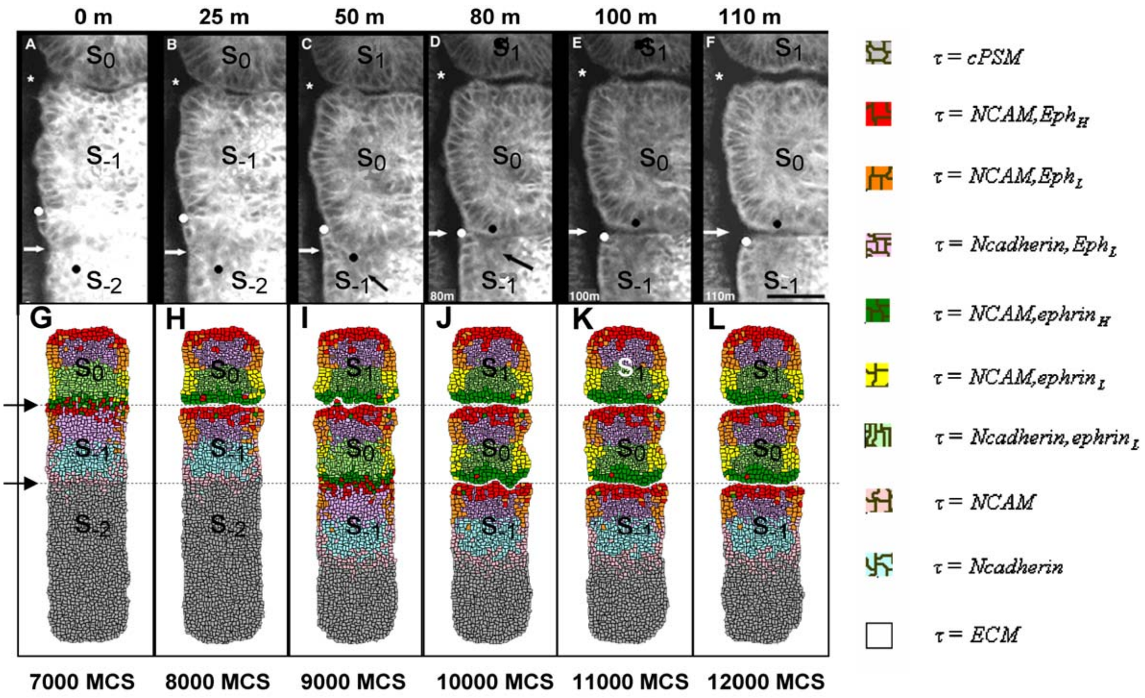

Figure 8 Simulated and experimental somite-segmentation dynamics. (A–F) Confocal time-lapse images of vitally-stained tissue (from Kulesa and Fraser, 2002; reprinted with permission from AAAS). White arrows indicate intersomitic boundaries. The white and black dots label cells which cross the presumptive boundary. (G–L) Simulated somite segmentation with preferential adhesion between PSM and ECM and AP indeterminacy in somite differentiation. Black arrows (➔) mark intersomitic boundaries. Scale bar in experimental image F is 50 μm and $J_{rHH} = J_{rHL} = 3J_{rLL} = 80$. See color insert.

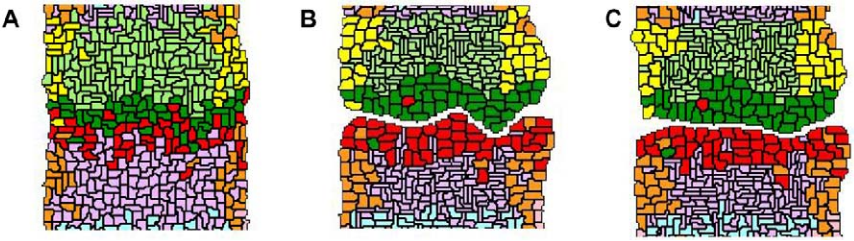

Figure 9 Detail of boundary crossing of misdifferentiated cells from Fig. 8. (A) 9000 MCS. (B) 10,000 MCS. (C) 11,000 MCS. See color insert.

The details of the retraction differ somewhat from some experiments, with the wings of the W retracting but not folding inwards, as is seen in other experiments.

Because biological signaling is noisy, some cells will slightly misread the somitic clock during determination. As a result they will differentiate inappropriately for their location (*misdifferentiate*). Such misdifferentiation occurs primarily near compartment boundaries, where cells are near a biological threshold between differentiation states and a small amount of noise can throw the switch. Initial fuzzy compartment boundaries followed by boundary refinement are common during development. Kulesa and Fraser may have observed such effects in their movies, where cells initially on one side of the presumptive intersomitic boundary cross the boundary during segmentation (Kulesa and Fraser, 2002). The white and black dots in Figs. 8A–8F denote cells that are not simply carried along as the W extends and retracts, but actually cross from S_{-2} to S_{-1} and from S_{-1} to S_{-2}, respectively.

Our simulations show that adhesion-based cell sorting provides a viable mechanism for the correction of minor errors in differentiation. In our simulation in Figs. 8G–8L, we have allowed for misdifferentiation by having each cell read its AP position with a small error chosen from a Gaussian distribution with a width δ of 14 microns when it selects its type, making the initial compartment boundaries blurry. Thus, some cells in S_{0P} assume types (red) that should be in S_{-1A}, etc., ... If we follow these red cells during their subsequent time evolution (Figs. 9A–9C), we find that adhesion-driven cell sorting causes most (though not all) of them to cross the compartment boundary into the correct compartment, in a manner identical to that in the experiments. The greater the cell motility compared to the rate of furrow formation, the more complete the correction will be.

VII. Conclusion

In this paper, we have shown that multilevel homotypic adhesion and heterotypic repulsion can reproduce many of the phenomena of normal somite segmentation, including the ball-and-socket dynamics and compartment crossing by misdifferentiated cells. Simulated N-cadherin knockouts produce an intrasomitic furrow as observed in experiments.

Acknowledgments

We acknowledge support from Grants NIGMS 1R01 GM076692-01, NSF IBN-0083653 and NASA NAG 2-1619, and from the Biocomplexity Institute, the Office of the Vice President for Research and the College of Arts and Sciences at Indiana University Bloomington. We thank Dr. A. W. Neff of the Indiana University School of Medicine for his critical reading of the manuscript, and Dr. O. Pourquié and his group members and Dr. P. Kulesa, all at the Stowers Institute for Medical Research, for helpful discussions. Y. Z. thanks the Department of Physics of Indiana University Bloomington for financial support.

Introduction to Appendices

To facilitate the validation, extension and reuse of complex multicell models, the interagency IMAG Consortium, which includes the NIGMS which primarily funded this research, has requested that all publications include source listings and related materials sufficient to allow a reader to duplicate and verify the primary computational claims made. In support of this goal, we provide in the following appendices, documented code which should suffice to replicate our results.

Running a GGH simulation using CompuCell3D requires initial installation of the CompuCell3D package (from www.compucell3d.org), followed by loading the appropriate CC3D ML and/or Python scripts into the CompuCell3D Player. The somitogenesis simulations we present in this paper employ both CC3D ML and Python scripts. The CC3D ML file contains the basic description of GGH parameters, such as lattice size, simulation duration in Monte Carlo Steps, Intrinsic Cell Motility (*Temperature*), definitions of adhesion energies between different types of cells, instructions to load Volume and Surface energy terms (the parameters of which Python script will initialize) and modules tracking cells' centers of mass and a Cell Lattice initialization routine (UniformInitializer).

The two Python scripts include one, somite.py that sets up the simulation (instantiating C^{++} and Python objects and implementing the main GGH loop) and a script that contains implementations of *Steppables* (somiteSteppables.py). Steppables define and manage transitions between cell types (SomiteMaskSteppable) and changes in Volume and Surface parameters (SomiteVolumeSurfaceSteppable).

The listings include brief comments explaining the significance of the code modules and indicating how to change the parameters to implement the different simulations in this paper. For convenience, complete files for implementing the somitogenesis simulations are also available for download from http://compucell3d.org/Models/somiteSimulationFiles.tgz.

Appendix A. Python Code to Execute Somitogenesis Simulations (somite.py)

```
def mainfcn():
   ### Initialization section
   import sys
   from os import environ
   import string
   sys.path.append(environ["PYTHON_MODULE_PATH"])

   import SystemUtils
   SystemUtils.setSwigPaths()
   SystemUtils.initializeSystemResources()

   import CompuCell
   import PlayerPython
   # This function wraps up the plugin initialization code.
   CompuCell.initializePlugins()

   # Create a Simulator. This section returns a Python object
   # that wraps Simulator.
   sim = CompuCell.Simulator()

   simthread=PlayerPython.getSimthreadBasePtr();
   simthread.setSimulator(sim)
   simulationFileName=simthread.getSimulationFileName()

   # Add the Python-specific extensions.
   reg = sim.getClassRegistry()

   CompuCell.parseConfig(simulationFileName, sim)
   ### End of Initialization section.

   ##############################
   # Registering objects that will allow adding
   # cell attributes during simulation runtime.
   # For more information see CompuCell3D Python Scripting
   # tutorials at www.compucell3d.org
   from CompuCell import PyAttributeAdder
   from PyListAdder import ListAdder
   from sys import getrefcount
   adder=PyAttributeAdder()
   adder.registerRefChecker(getrefcount)
   listAdder=ListAdder()
   adder.registerAdder(listAdder)
   potts=sim.getPotts()
   potts.registerAttributeAdder(adder.getPyAttributeAdderPtr())
```

```
### Further initialization of Player and CompuCell3D C++
# code.
dim=sim.getPotts().getCellFieldG().getDim()
# After all CC3D ML steppables and plugins have been loaded
# we call extraInit to complete the initialization.
sim.extraInit()
simthread.preStartInit()
sim.start()
simthread.postStartInit()
screenUpdateFrequency=simthread.getScreenUpdateFrequency()

### Instantiating and registering Python-based plugins.
# Notice that all Python steppables are registered
# with steppableRegistry,
# e.g., steppableRegistry.registerSteppable
# (somiteMaskSteppable) from PySteppables import
# SteppableRegistry

steppableRegistry=SteppableRegistry()

from somiteSteppables import SomiteMaskSteppable
somiteMaskSteppable=SomiteMaskSteppable
(_simulator=sim,_frequency=10)
# Sweep as [T0, T0+T, T0+2T],
# T0=2000, 3000, 400, 5000,6000;
#  T=1000,2000, 3000, 4000, 5000;
# Here you specify the MC steps at which transitions
# take place.
transitionStepsList=[5000,7000,9000]
somiteMaskSteppable.setTransitionStepsList
(transitionStepsList)
somiteMaskSteppable.blur(0.0)
steppableRegistry.registerSteppable(somiteMaskSteppable)

from somiteSteppables import SomiteVolumeSurfaceSteppable
somiteVolumeSurfaceSteppable=SomiteVolumeSurfaceSteppable(\
_simulator=sim,_frequency=100)
# Here you specify the max MCS at which changes take place -
# this way you do not execute code in this steppable after
# all transitions have taken place.
somiteVolumeSurfaceSteppable.setMaxTransitionStep(\
somiteMaskSteppable.getMaxTransitionStep())
steppableRegistry.registerSteppable
(somiteVolumeSurfaceSteppable)

steppableRegistry.init(sim)
```

```
   steppableRegistry.start()

   ### Main GGH algorithm loop.
   for i in range(sim.getNumSteps()):
      sim.step(i)
      steppableRegistry.step(i)
      if not i % screenUpdateFrequency:
         simthread.loopWork(i)
         simthread.loopWorkPostEvent(i)
   sim.finish()
   steppableRegistry.finish()
mainfcn()
```

Appendix B. CC3D ML Code to Execute Somitogenesis Simulations (somite.xml)

```
--<CompuCell3D>
 <!-- This section defines the basic parameters of the
  GGH model.-->
 <Potts>
   <Dimensions x="170" y="1" z="450"/>
   <Anneal>1</Anneal>
   <Steps>100</Steps>
   <Temperature>100</Temperature>
   <Flip2DimRatio>1</Flip2DimRatio>
 </Potts>

<!-- Each CompuCell3D CC3D ML file must include a list
of all cell types used in the simulation.-->
<Plugin Name="CellType">
   <CellType TypeName="Medium" TypeId="0"/>
   <CellType TypeName="ncam" TypeId="1"/>
   <CellType TypeName="ncad" TypeId="2" />
   <CellType TypeName="eph-rec" TypeId="3"/>
   <CellType TypeName="eph-rec-ncam" TypeId="4" />
   <CellType TypeName="eph-rec-ncad" TypeId="5" />
   <CellType TypeName="eph-lig-ncam" TypeId="6" />
   <CellType TypeName="eph-lig-ncad" TypeId="7" />
   <CellType TypeName="eph-lig" TypeId="8" />
   <CellType TypeName="psm" TypeId="9" />
</Plugin>

<!-- This plugin tells the player which lattice
projection should be displayed at the start.-->
<Plugin Name="PlayerSettings">
   <Project2D XZProj="1"/>
   <InitialProjection Projection="xz"/>
</Plugin>

<!-- Additional initialization of volume and surface target
values and lambdas is required. This is done in the
Python scripts.-->
<Plugin Name="VolumeLocalFlex"/>
<Plugin Name="SurfaceLocalFlex"/>

<!-- List of contact energies between different cell types.-->
<Plugin Name="Contact">
    <Energy Type1="Medium" Type2="Medium">0</Energy>
    <Energy Type1="Medium" Type2="ncam">15</Energy>
    <Energy Type1="Medium" Type2="ncad">15</Energy>
    <Energy Type1="Medium" Type2="eph-rec">15</Energy>
    <Energy Type1="Medium" Type2="eph-rec-ncam">15</Energy>
    <Energy Type1="Medium" Type2="eph-rec-ncad">15</Energy>
    <Energy Type1="Medium" Type2="eph-lig-ncam">15</Energy>
    <Energy Type1="Medium" Type2="eph-lig-ncad">15</Energy>
    <Energy Type1="Medium" Type2="eph-lig">15</Energy>
    <Energy Type1="Medium" Type2="psm">15</Energy>
    <Energy Type1="ncam" Type2="ncam">-20.25</Energy>
    <Energy Type1="ncam" Type2="ncad">-24</Energy>
    <Energy Type1="ncam" Type2="eph-rec">-20.25</Energy>
    <Energy Type1="ncam" Type2="eph-rec-ncam">-20.25</Energy>
    <Energy Type1="ncam" Type2="eph-rec-ncad">-24</Energy>
    <Energy Type1="ncam" Type2="eph-lig-ncam">-20.25
    </Energy>
    <Energy Type1="ncam" Type2="eph-lig-ncad">-24</Energy>
    <Energy Type1="ncam" Type2="eph-lig">-20.25</Energy>
    <Energy Type1="ncam" Type2="psm">-20.25</Energy>
    <Energy Type1="ncad" Type2="ncad">-38.44</Energy>
    <Energy Type1="ncad" Type2="eph-rec">-24</Energy>
    <Energy Type1="ncad" Type2="eph-rec-ncam">-24</Energy>
    <Energy Type1="ncad" Type2="eph-rec-ncad">-38.44</Energy>
    <Energy Type1="ncad" Type2="eph-lig-ncam">-24</Energy>
    <Energy Type1="ncad" Type2="eph-lig-ncad">-38.44</Energy>
    <Energy Type1="ncad" Type2="eph-lig">-24</Energy>
    <Energy Type1="ncad" Type2="psm">-38.44</Energy>
    <Energy Type1="eph-rec" Type2="eph-rec">-20.25</Energy>
    <Energy Type1="eph-rec" Type2="eph-rec-ncam">-20.25
    </Energy>
    <Energy Type1="eph-rec" Type2="eph-rec-ncad">-24</Energy>
    <Energy Type1="eph-rec" Type2="eph-lig-ncam">-20.25</Energy>
    <Energy Type1="eph-rec" Type2="eph-lig-ncad">-24.0</Energy>
    <Energy Type1="eph-rec" Type2="eph-lig">-20.25</Energy>
    <Energy Type1="eph-rec" Type2="psm">-20.25</Energy>
    <Energy Type1="eph-rec-ncam" Type2="eph-rec-ncam">-20.25
    </Energy>
    <Energy Type1="eph-rec-ncam" Type2="eph-rec-ncad">-24
    </Energy>
    <Energy Type1="eph-rec-ncam" Type2="eph-lig-ncam">-20.25
    </Energy>
    <Energy Type1="eph-rec-ncam" Type2="eph-lig-ncad">-24.0
    </Energy>
    <Energy Type1="eph-rec-ncam" Type2="eph-lig">-20.25
    </Energy>
    <Energy Type1="eph-rec-ncam" Type2="psm">-20.25</Energy>
    <Energy Type1="eph-rec-ncad" Type2="eph-rec-ncad">-38.44
    </Energy>
    <Energy Type1="eph-rec-ncad" Type2="eph-lig-ncam">-24
    </Energy>
    <Energy Type1="eph-rec-ncad" Type2="eph-lig-ncad">-38.44
    </Energy>
    <Energy Type1="eph-rec-ncad" Type2="eph-lig">-24.0</Energy>
    <Energy Type1="eph-rec-ncad" Type2="psm">-38.44</Energy>
    <Energy Type1="eph-lig-ncam" Type2="eph-lig-ncam">-20.25
    </Energy>
    <Energy Type1="eph-lig-ncam" Type2="eph-lig-ncad">-24
    </Energy>
    <Energy Type1="eph-lig-ncam" Type2="eph-lig">-20.25
    </Energy>
    <Energy Type1="eph-lig-ncam" Type2="psm">-20.25</Energy>
    <Energy Type1="eph-lig-ncad" Type2="eph-lig-ncad">-38.44
    </Energy>
    <Energy Type1="eph-lig-ncad" Type2="eph-lig">-24</Energy>
    <Energy Type1="eph-lig-ncad" Type2="psm">-38.44</Energy>
    <Energy Type1="eph-lig" Type2="eph-lig">-20.25</Energy>
    <Energy Type1="eph-lig" Type2="psm">-20.25</Energy>
    <Energy Type1="psm" Type2="psm">-20.25</Energy>
    <Depth>2.3</Depth>
</Plugin>

<!-- This plugin tracks the center of mass of each cell
and is necessary for laying out the prepattern of cadherin
expression.-->
<Plugin Name="CenterOfMass"/>

<!-- UniformInitializer lays out the initial pattern of cells.
In our case, a rectangular block corresponding to the
presomitic mesoderm (psm).-->
<Steppable Type="UniformInitializer">
    <Region>
        <BoxMin x="35" y="0" z="30"/>
        <BoxMax x="135" y="1" z="430"/>
        <Gap>0</Gap>
        <Width>5</Width>
       <Types>psm</Types>
    </Region>
</Steppable>
</CompuCell3D>
```

Appendix C. Python Steppables for Somitogenesis Simulations (somiteSteppables.py)

```
from PySteppables import *
import CompuCell
import sys
import CompuCell
import random

# The function forEachCellInInventory takes as arguments
# the inventory of cells and a function that will operate on a
# single cell. It runs singleCellOperation on each cell from
# the cell inventory.
def forEachCellInInventory(inventory,singleCellOperation):
   invItr=CompuCell.STLPyIteratorCINV()
   invItr.initialize(inventory.getContainer())
   invItr.setToBegin()
   cell=invItr.getCurrentRef()
   while (not invItr.isEnd()):
      cell=invItr.getCurrentRef()
      singleCellOperation(cell)
      invItr.next()

# TypeTransition is a class that describes a
# cell-type transition (self.typeIdSource, self.typeIdTarget)
# and at what time (in MCS) the transition should take place.
class TypeTransition:
   def __init__(self,_typeIdSource,_typeIdTarget,_step):
      self.typeIdSource=_typeIdSource
      self.typeIdTarget=_typeIdTarget
      self.step=_step

# The SomiteMaskSteppable defines a set of masks that describe
# cell types. Overlaying a mask is equivalent to defining new
# type transitions for a given cell. Masks are overlaid before
# simulation begins, which is why we can use fixed mask
# coordinates (e.g., self.x\_rect\_low=45). As discussed
# in the paper, the development of the prepattern of cadherin
# expression can be thought of as a series of cell-type
# transitions.
class SomiteMaskSteppable(SteppablePy):
   def __init__(self,_simulator,_frequency=1):
      SteppablePy.__init__(self,_frequency)
      self.simulator=_simulator
      self.cellFieldG=self.simulator.getPotts().getCellFieldG()
      self.dim=self.cellFieldG.getDim()
      self.inventory=self.simulator.getPotts().
      getCellInventory()
      self.typeNameTable={"Medium":0,"AL":1,"RH":2,"3":3,"4":4,
      "5":5,\
      "6":6,"7":7,"8":8,"PSM"':9}

      #base mask parameters
      self.z_base_low=30
      self.z_base_size=100
      #mask rectangle parameters
      self.x_rect_low=45
      self.z_rect_low=40
      self.x_rect_size=80
      self.z_rect_size=80
      #mask0 parameters
      self.z_low_mask0=30
      self.z_size_mask0=10
      #mask1 parameters
      self.z_low_mask1=40
      self.z_size_mask1=40
      #mask2 parameters
      self.z_low_mask2=80
      self.z_size_mask2=40
      #mask3 parameters
      self.z_low_mask3=120
      self.z_size_mask3=10
      # The above determine the shifts in the masks'
        z_low positions -
      # They should be applied to all masks in the correct
      # order.
      self.shift=100
      self.maskMargin=10
      self.stepForTransition=0
      self.transitionStepsList=[]
      self.maxTransitionStep=0
      self.sigmaBlur=0.0

   def getMaxTransitionStep(self):
      return self.maxTransitionStep
```

```
def setTransitionStepsList(self,_transitionStepsList):
   self.transitionStepsList=_transitionStepsList
   self.maxTransitionStep=max(self.transitionStepsList)

# Whenever we overlay a mask, we attach to each cell a
# transition from its current type to a target type.
# Later, when we make the transition, we simply read from
# the attached list of transitions and perform those for
# which the current time (in MCS) matches the times defined
# in the transition-description list.

# The functions below implement a series of masks that are
# used to construct the prepattern.
def maskBase(self,cell,xCM,yCM,zCM,attrib):
   if (zCM >= self.z_base_low and zCM<self.z_base_low
       +self.z_base_size):
      sourceId=cell.type
      cell.type=self.typeNameTable["AL"]
      # To record a type transition.
      attrib.append(TypeTransition(sourceId,cell.type,
                    self.stepForTransition))

def maskRectangle(self,cell,xCM,yCM,zCM,attrib):
   if(xCM>=self.x_rect_low and xCM <self.x_rect_low
      +self.x_rect_size\
   and zCM >= self.z_rect_low and zCM<self.z_rect_low
       +self.z_rect_size and\
   cell.type==self.typeNameTable["AL"]):
      sourceId=cell.type
      cell.type=self.typeNameTable["RH"]
      attrib.append(TypeTransition(sourceId,cell.type,
      self.stepForTransition))

def mask0(self,cell,xCM,yCM,zCM,attrib):
   if(zCM>=self.z_low_mask0 and zCM <self.z_low_mask0+
      self.z_size_mask0):
     if(cell.type==self.typeNameTable["AL"]):
        sourceId=cell.type
        cell.type=self.typeNameTable["3"]
        attrib.append(TypeTransition(sourceId,cell.type,\
        self.stepForTransition))

def mask1(self,cell,xCM,yCM,zCM,attrib):
   if(zCM>=self.z_low_mask1 and zCM <self.z_low_mask1+
      self.z_size_mask1):
     if(cell.type==self.typeNameTable["AL"]):
        sourceId=cell.type
        cell.type=self.typeNameTable["4"]
        attrib.append(TypeTransition(sourceId,cell.type,\
        self.stepForTransition))
     elif(cell.type==self.typeNameTable["RH"]):
        sourceId=cell.type
        cell.type=self.typeNameTable["5"]
        attrib.append(TypeTransition(sourceId,cell.type,\
        self.stepForTransition))

def mask2(self,cell,xCM,yCM,zCM,attrib):
   if(zCM>=self.z_low_mask2-self.maskMargin and zCM \
   <self.z_low_mask2+self.z_size_mask2):
      if(cell.type==self.typeNameTable["AL"]):
         sourceId=cell.type
         cell.type=self.typeNameTable["6"]
         attrib.append(TypeTransition(sourceId,cell.type,\
         self.stepForTransition))
      elif(cell.type==self.typeNameTable["RH"]):
         sourceId=cell.type
         cell.type=self.typeNameTable["7"]
         attrib.append(TypeTransition(sourceId,cell.type,\
         self.stepForTransition))

def mask3(self,cell,xCM,yCM,zCM,attrib):
   if(zCM>=self.z_low_mask3 and zCM <self.z_low_mask3+
      self.z_size_mask3):
      sourceId=cell.type
      cell.type=self.typeNameTable["8"]
      attrib.append(TypeTransition(sourceId,cell.type,
      self.stepForTransition))

def maskPSM(self,cell,xCM,yCM,zCM,attrib):
   cell.type=self.typeNameTable["PSM"]

def blur(self,_sigmaBlur):
self.sigmaBlur=_sigmaBlur

# The imposeMask function is a Closure that takes as
# its first argument mask and returns a function
# that operates on a single cell, making use
# of the mask object. For more information on Closures
# in Python please consult, e.g., the "Python Cookbook,"
# or search using a web search-engine, using "python Closure"
# as keywords. Notice that as a result (return
# imposeMaskForSingleCell) we obtain a function that operates
# on a single cell exactly as required by the
# forEachCellInInventory algorithm.
```

```
def imposeMask(self,mask):
   def imposeMaskForSingleCell(cell):
      xCM=cell.xCM/float(cell.volume)
      yCM=cell.yCM/float(cell.volume)
      zCM=cell.zCM/float(cell.volume)
      pyAttrib=CompuCell.getPyAttrib(cell)
      mask(cell,xCM,yCM,zCM,pyAttrib)
   return imposeMaskForSingleCell

# Overlaying masks can be coded very elegantly in just
# two lines making use of the imposeMask Closure and the
# forEachCellInInventory algorithm.
def overlayMasks(self,mask):
   imposeMaskFunction=self.imposeMask(mask)
   forEachCellInInventory(inventory=self.inventory,\
   singleCellOperation=imposeMaskFunction)

# The singleCellTransition Closure returns a function
# that operates on a single cell and implements a type
# transition for a single cell. We had to use a Closure
# because we are using an extra argument _mcs in addition
# to the cell variable.
def singleCellTransition(self,_mcs):
   def transition(cell):
      pyAttrib=CompuCell.getPyAttrib(cell)
      attribSize=len(pyAttrib)
      if(attribSize>1):
         for i in xrange(1,attribSize):
            # The transition will take place if and only if
            # the current time (in MCS) (_mcs)
            # matches the step variable defined in the
            # transition object.
            if(pyAttrib[i].step==_mcs):
               cell.type=pyAttrib[i].typeIdTarget
   return transition

# The doTransitions function iterates over each cell
# and makes type transitions as necessary.
def doTransitions(self,_mcs):
   singleCellTransitionFunction=
   self.singleCellTransition(_mcs)
   forEachCellInInventory(inventory=self.inventory,\
   singleCellOperation=singleCellTransitionFunction)

# The following implements the misdifferentiation of cells
# due to inaccurate positional information.
# The doBlurSingleCell Closure returns a function operating
# on a single cell object that calculates the center of mass
# cCM of a given cell, adds a random vector V to it
# (the coordinates of which are chosen from a Gaussian
# distribution) and changes the type of the current cell
# to the type of a cell located at cCM+V (or leaves cell
# type untouched if the cell at cCM+V happens to be medium).

def doBlurSingleCell(self,_mcs):
   def blurFunction(cell):
      xCM=cell.xCM/float(cell.volume)
      yCM=cell.yCM/float(cell.volume)
      zCM=cell.zCM/float(cell.volume)
    if self.sigmaBlur!=0.0:
       # print "Will do the bluring ",
         random.gauss(0.0,self.sigmaBlur)
       xCM+=random.gauss(0.0,self.sigmaBlur)
       yCM+=random.gauss(0.0,self.sigmaBlur)
       zCM+=random.gauss(0.0,self.sigmaBlur)
      pt=CompuCell.Point3D()
    pt.x=int(xCM)
    pt.y=int(yCM)
    pt.z=int(zCM)
    neighborCell=self.cellFieldG.get(pt)
    if neighborCell:
         cell.type=neighborCell.type
    return blurFunction

# Iterate over each cell and apply the blur
# (misdifferentiation).
def doBlur(self,_mcs):
   doBlurSingleCellFunction=self.doBlurSingleCell(_mcs)
   forEachCellInInventory(inventory=self.inventory,\
   singleCellOperation=doBlurSingleCellFunction)

# This function is run before the simulation begins.
# It lays out the cadherin prepattern, i.e., it initializes
# a set of potential type transitions for each cell.
def start(self):

   if(len(self.transitionStepsList)<3):
      print "You need to provide list with MC steps \
      at which spin reassignment takes place"
      sys.exit()
   # Initialize the anterior compartment (low z coordinates)
   # initial prepattern.
   self.stepForTransition=self.transitionStepsList[0]
   self.overlayMasks(self.maskBase)
```

```
      self.overlayMasks(self.maskRectangle)
      self.overlayMasks(self.mask0)
      self.overlayMasks(self.mask1)

      # Initialize transitions for anterior (low z) middle
      # (medium z) and posterior (high z) somite cells.
      # Notice that we go from low values of z to higher
      # values of z by manipulating class variables
      # such as self.z_base_low and self.z_low_mask0 and
      # shifting them for each of the transition steps.
      for i in xrange(1,3):
         self.stepForTransition=self.transitionStepsList[i]
         self.overlayMasks(self.mask2)
         self.overlayMasks(self.mask3)
         # z_low for mask 2 and 3 is shifted after the call
         # to overlay masks.
         self.z_low_mask2+=self.shift
         self.z_low_mask3+=self.shift
         self.z_base_low+=self.shift
         self.z_rect_low+=self.shift
         self.overlayMasks(self.maskBase)
         self.overlayMasks(self.maskRectangle)
         self.z_low_mask0+=self.shift
         self.z_low_mask1+=self.shift
         self.overlayMasks(self.mask0)
         self.overlayMasks(self.mask1)
      # Because the overlayMasks function has the side effect
      # of changing cell types, we need to reset the cell
      # types after we impose the masks. At the beginning
      # of the simulation all the cells are of type PSM
      # (presomitic mesoderm).
      self.overlayMasks(self.maskPSM)
   # This function is run every 10 MCS -
   # see somiteMaskSteppable=
   # SomiteMaskSteppable(_simulator=sim,_frequency=10)in
   # somite.py.
   def step(self,mcs):
      if(mcs<=self.maxTransitionStep):
         if mcs in self.transitionStepsList:
            self.doTransitions(mcs)
            self.doBlur(mcs)
            return
      return

   # SomiteVolumeSurfaceSteppable is responsible for
   # periodically assigning new volume and surface energy
   # parameters. Those parameters are local to each cell,
   # so when cells change types these parameters need to be
   # updated as well. We have hard-coded the parameters
   # (targerVolume, lambda) (targetSurface, lambda).
   class SomiteVolumeSurfaceSteppable(SteppablePy):
      def __init__(self,_simulator,_frequency=1):
         SteppablePy.__init__(self,_frequency)
         self.simulator=_simulator
         self.cellFieldG=self.simulator.getPotts().
         getCellFieldG()
         self.dim=self.cellFieldG.getDim()
         self.inventory=self.simulator.getPotts().
         getCellInventory()
         self.maxTransitionStep=0
         #format
         #type:[targetVolume,lambda]
         #example 2:[25.0,20.0]
         self.typeVolumeParamMap={1:[25.0,20.0], 2:[25.0,20.0],
         3:[36.0,20.0],\
         4:[36.0,20.0],5:[16.0,20.0],6:[36.0,20.0],
         7:[16.0,20.0],
         8:[36.0,20.0],9:[25.0,20.0]}
         #format
         #type:[targetSurface,lambda]
         #example 2:[20.0,20.0]

   self.typeSurfaceParamMap={1:[20.0,20.0],2:[20.0,20.0],
   3:[24.0,20.0],4:[24.0,20.0],\
   5:[16.0,20.0],6:[24.0,20.0],7:[16.0,20.0],
   8:[24.0,20.0],9:[20.0,20.0]}


    def setMaxTransitionStep(self,_maxTransitionStep):
        self.maxTransitionStep=_maxTransitionStep

     # At the beginning of the simulation all cells have
     # these volume parameters.
     def volumeInitSet(self,cell):
        cell.targetVolume=25.0
        cell.lambdaVolume=20.0
     # At the beginning of the simulation all cells have
     # these surface parameters.
     def surfaceInitSet(self,cell):
        cell.targetSurface=20.0
        cell.lambdaSurface=20.0

   # At later stages we will change the volume and
   # surface parameters using
```

```
   # self.typeVolumeParamMap and self.typeSurfaceParamMap.
   def volumeParamSet(self,cell):
      par=self.typeVolumeParamMap[cell.type]
      cell.targetVolume=par[0]
      cell.lambdaVolume=par[1]
   def surfaceParamSet(self,cell):
      par=self.typeSurfaceParamMap[cell.type]
      cell.targetSurface=par[0]
      cell.lambdaSurface=par[1]

   def setParameters(self,paramSetFcn):
      forEachCellInInventory(inventory=self.inventory,\
      singleCellOperation=paramSetFcn)

   def start(self):
      self.setParameters(self.volumeInitSet)
      self.setParameters(self.surfaceInitSet)

   # This function is run every 100 MCS-
   # see somiteVolumeSurfaceSteppable=
   # SomiteVolumeSurfaceSteppable(_simulator=sim,_frequency
   # =100) in somte.py. Notice that we run this function
   # whether a transition took place or not, which is redundant,
   # but makes the code easier to read.
   def step(self,mcs):
      if(mcs<=self.maxTransitionStep):
         self.setParameters(self.volumeParamSet)
         self.setParameters(self.surfaceParamSet)
      return
```

References

Afonin, B., Ho, M., Gustin, J. K., Meloty-Kapella, C., and Domingo, C. R. (2006). Cell behaviors associated with somite segmentation and rotation in Xenopus laevis. *Dev. Dyn.* **235**, 3268–3279.

Aulehla, A., and Herrmann, B. G. (2004). Segmentation in vertebrates: Clock and gradient finally joined. *Genes Dev.* **18**, 2060–2067.

Aulehla, A., and Pourquié, O. (2006). On periodicity and directionality of somitogenesis. *Anat. Embryol.* **211**, S3–S8.

Aulehla, A., Wehrle, C., Brand-Saberi, B., Kemler, R., Gossler, A., Kanzler, B., and Herrmann, B. G. (2003). Wnt3a plays a major role in the segmentation clock controlling somitogenesis. *Dev. Cell* **4**, 395–406.

Baker, R. K., and Antin, P. B. (2003). Ephs and ephrins during early stages of chick embryogenesis. *Dev. Dynam.* **228**, 128–142.

Baker, R. E., Schnell, S., and Maini, P. K. (2003). Formation of vertebral precursors: Past models and future predictions. *J. Theor. Med.* **5**, 23–35.

Baker, R. E., Schnell, S., and Maini, P. K. (2006). A clock and wavefront mechanism for somite formation. *Dev. Biol.* **293**, 116–126.

Barrios, A., Poole, R. J., Durbin, L., Brennan, C., Holder, N., and Wilson, S. W. (2003). Eph/Ephrin signaling regulates the mesenchymal-to-epithelial transition of the paraxial mesoderm during somite morphogenesis. *Curr. Biol.* **13**, 1571–1582.

Bergemann, A. D., Cheng, H. J., Brambilla, R., Klein, R., and Flanagan, J. G. (1995). ELF-2, a new member of the Eph ligand family, is segmentally expressed in mouse embryos in the region of the hindbrain and newly forming somites. *Mol. Cell Biol.* **15**, 4921–4929.

Chaturvedi, R., Huang, C., Kazmierczak, B., Schneider, T., Izaguirre, J. A., Glimm, T., Hentschel, H. G. E., Glazier, J. A., Newman, S. A., and Alber, M. S. (2005). On multiscale approaches to three-dimensional modeling of morphogenesis. *J. R. Soc. Interface* **2**, 237–253.

Collier, J. R., McInerney, D., Schnell, S., Maini, P. K., Gavaghan, D. J., Houston, P., and Stern, C. D. (2000). A cell cycle model for somitogenesis: Mathematical formulation and numerical simulation. *J. Theor. Biol.* **207**, 305–316.

Cooke, J., and Zeeman, E. C. (1976). A clock and wavefront model for control of the number of repeated structures during animal morphogenesis. *J. Theor. Biol.* **58**, 455–476.

Cooke, J. E., Kemp, H. A., and Moens, C. B. (2005). EDhA4 is required for cell adhesion and rhombomere-boundary formation in the zebrafish. *Curr. Biol.* **15**, 536–542.

De Bellard, M. E., Ching, W., Gossler, A., and Bronner-Fraser, M. (2002). Disruption of segmental neural crest migration and ephrin expression in delta-1 null mice. *Dev. Biol.* **249**, 121–130.

Dequeant, M. L., Glynn, E., Gaudenz, K., Wahl, M., Chen, J., Mushegian, A., and Pourquié, O. (2006). A complex oscillating network of signaling genes underlies the mouse segmentation clock. *Science* **314**, 1595–1598.

Diez del Corral, R., Olivera-Martinez, I., Goriely, A., Gale, E., Maden, M., and Storey, K. (2003). Opposing FGF and retinoid pathways control ventral neural pattern, neuronal differentiation, and segmentation during body axis extension. *Neuron* **40**, 65–79.

Duband, J. L., Dufour, S., Hatta, K., Takeichi, M., Edelman, G. M., and Thiery, J. P. (1987). Adhesion molecules during somitogenesis in the avian embryo. *J. Cell Biol.* **104**, 1361–1374.

Dubrulle, J., and Pourquié, O. (2002). From head to tail: Links between the segmentation clock and antero-posterior patterning of the embryo. *Curr. Opin. Genet. Dev.* **12**, 519–523.

Dubrulle, J., and Pourquié, O. (2004). Coupling segmentation to axis formation. *Development* **131**, 5783–5793.

Dubrulle, J., McGrew, M. J., and Pourquié, O. (2001). FGF signaling controls somite boundary position and regulates segmentation clock control of spatiotemporal Hox gene activation. *Cell* **106**, 219–232.

Durbin, L., Brennan, C., Shiomi, K., Cooke, J., Barrios, A., Shanmugalingam, S., Guthrie, B., Lindberg, R., and Holder, N. (1998). Eph signaling is required for segmentation and differentiation of the somites. *Genes Dev.* **12**, 3096–3109.

Foty, R. A., and Steinberg, M. S. (2005). The differential adhesion hypothesis: A direct evaluation. *Dev. Biol.* **278**, 255–263.

Glazier, J. A., and Graner, F. (1993). Simulation of the differential adhesion driven rearrangement of biological cells. *Phys. Rev. E* **47**, 2128–2154.

Goldbeter, A., Gonze, D., and Pourquié, O. (2007). Sharp developmental thresholds defined through bistability by antagonistic gradients of retinoic acid and FGF signaling. *Dev. Dyn.* **236**, 1495–1508.

Gossler, A., and Hrabe de Angelis, M. (1998). Somitogenesis. *Curr. Top. Dev. Biol.* **38**, 225–287.

Graner, F., and Glazier, J. A. (1992). Simulation of biological cell sorting using a two-dimensional extended Potts model. *Phys. Rev. Lett.* **69**, 2013–2016.

Grima, R., and Schnell, S. (2007). Can tissue surface tension drive somite formation? *Dev. Biol.* **307**, 248–257.

Harbott, L. K., and Nobes, C. D. (2005). A key role for Abl family kinases in EphA receptor-mediated growth cone. *Mol. Cell. Neurosci.* **30**, 1–11.

Henry, C. A., Hall, L. A., Burr Hille, M., Solnica-Krezel, L., and Cooper, M. S. (2000). Somites in zebrafish doubly mutant for knypek and trilobite form without internal mesenchymal cells or compaction. *Curr. Biol.* **10**, 1063–1066.

Horikawa, K., Radice, G., Takeichi, M., and Chisaka, O. (1999). Adhesive subdivisions intrinsic to the epithelial somites. *Dev. Biol.* **215**, 182–189.

Horikawa, K., Ishimatsu, K., Yoshimoto, E., Kondo, S., and Takeda, H. (2006). Noise-resistant and synchronized oscillation of the segmentation clock. *Nature* **441**, 719–723.

Jiang, Y. J., Aerne, B. L., Smithers, L., Haddon, C., Ish-Horowicz, D., and Lewis, J. (2000). Notch signaling and the synchronization of the somite segmentation clock. *Nature* **408**, 475–479.

Kasemeier-Kulesa, J. C., Bradley, R., Pasquale, E. B., Lefcort, F., and Kulesa, P. M. (2006). Eph/ephrins and N-cadherin coordinate to control the pattern of sympathetic ganglia. *Development* **133**, 4839–4847.

Keynes, R. J., and Stern, C. D. (1988). Mechanisms of vertebrate segmentation. *Development* **103**, 413–429.

Kimura, Y., Matsunami, H., Inoue, T., Shimamura, K., Uchida, N., Ueno, T., Miyazaki, T., and Takeichi, M. (1995). Cadherin-11 expressed in association with mesenchymal morphogenesis in the head, somite, and limb bud of early mouse embryos. *Dev. Biol.* **169**, 347–358.

Kulesa, P. M., and Fraser, S. E. (2002). Cell dynamics during somite boundary formation revealed by time-lapse analysis. *Science* **298**, 991–995.

Kulesa, P. M., Schnell, S., Rudloff, S., Baker, R. E., and Maini, P. K. (2007). From segment to somite: Segmentation to epithelialization analyzed within quantitative frameworks. *Dev. Dyn.* **236**, 1392–1402.

Linask, K. K., Ludwig, C., Han, M. D., Liu, X., Radice, G. L., and Knudsen, K. A. (1998). N-cadherin/catenin-mediated morphoregulation of somite formation. *Dev. Biol.* **202**, 85–102.

Marée, A. F., and Hogeweg, P. (2001). How amoeboids self-organize into a fruiting body: Multicellular coordination in Dictyostelium discoideum. *Proc. Natl. Acad. Sci. USA* **98**, 3879–3883.

Meinhardt, H. (1996). Models of biological pattern formation: Common mechanism in plant and animal development. *Int. J. Dev. Biol.* **40**, 123–134.

Mellitzer, G., Xu, Q., and Wilkinson, D. G. (1999). Eph receptors and ephrins restrict cell intermingling and communication. *Nature* **400**, 77–81.

Merks, R. M., and Glazier, J. A. (2006). Dynamic mechanisms of blood vessel growth. *Nonlinearity* **19**, C1–C10.

Merks, R. M., Brodsky, S. V., Goligorksy, M. S., Newman, S. A., and Glazier, J. A. (2006). Cell elongation is key to in silico replication of in vitro vasculogenesis and subsequent remodeling. *Dev. Biol.* **289**, 44–54.

Moreno, T. A., and Kintner, C. (2004). Regulation of segmental patterning by retinoic acid signaling during Xenopus somitogenesis. *Dev. Cell* **6**, 205–218.

Nakajima, Y., Morimoto, M., Takahashi, Y., Koseki, H., and Saga, Y. (2006). Identification of Epha4 enhancer required for segmental expression and the regulation by Mesp2. *Development* **133**, 2517–2525.

Nelson, W. J., and Nusse, R. (2004). Convergence of Wnt, beta-catenin, and cadherin pathways. *Science* **303**, 1483–1487.

Nieto, M. A., Gilardi-Hebenstreit, P., Charnay, P., and Wilkinson, D. G. (1992). A receptor protein tyrosine kinase implicated in the segmental patterning of the hindbrain and mesoderm. *Development* **116**, 1137–1150.

Ordahl, C. P. (1993). Myogenic lineages within the developing somite. *In* "Molecular Basis of Morphogenesis." (M. Bernfield, Ed.). Wiley, New York, pp. 165–176.

Palmeirim, I., Henrique, D., Ish-Horowicz, D., and Pourquié, O. (1997). Avian hairy gene expression identifies a molecular clock linked to vertebrate segmentation and somitogenesis. *Cell* **91**, 639–648.

Poliakov, A., Cotrina, M., and Wilkinson, D. G. (2004). Diverse roles of eph receptors and ephrins in the regulation of cell migration and tissue assembly. *Dev. Cell* **7**, 465–480.

Poplawski, N. J., Swat, M., Gens, J. S., and Glazier, J. A. (2007). Adhesion between cells, diffusion of growth factors, and elasticity of the AER produce the paddle shape of the chick limb. *Physica A* **373C**, 521–532.

Pourquié, O. (2004). The chick embryo: A leading model in somitogenesis studies. *Mech. Dev.* **121**, 1069–1079.

Pourquié, O., and Tam, P. P. (2001). A nomenclature for prospective somites and phases of cyclic gene expression in the presomitic mesoderm. *Dev. Cell* **1**, 619–620.

Primmett, D. R., Stern, C. D., and Keynes, R. J. (1988). Heat shock causes repeated segmental anomalies in the chick embryo. *Development* **104**, 331–339.

Primmett, D. R., Norris, W. E., Carlson, G. J., Keynes, R. J., and Stern, C. D. (1989). Periodic segmental anomalies induced by heat shock in the chick embryo are associated with the cell cycle. *Development* **105**, 119–130.

Radice, G. L., Rayburn, H., Matsunami, H., Knudsen, K. A., Takeichi, M., and Hynes, R. O. (1997). Developmental defects in mouse embryos lacking N-cadherin. *Dev. Biol.* **181**, 64–78.

Sato, Y., and Takahashi, Y. (2005). A novel signal induces a segmentation fissure by acting in a ventral-to-dorsal direction in the presomitic mesoderm. *Dev. Biol.* **282**, 183–191.

Sato, Y., Yasuda, K., and Takahashi, Y. (2002). Morphological boundary forms by a novel inductive event mediated by Lunatic fringe and Notch during somitic segmentation. *Development* **129**, 3633–3644.

Schnell, S., and Maini, P. K. (2000). Clock and induction model for somitogenesis. *Dev. Dynam.* **217**, 415–420.

Schnell, S., Maini, P. K., McInerney, D., Gavaghan, D. J., and Houston, P. (2002). Models for pattern formation in somitogenesis: A marriage of cellular and molecular biology. *C. R. Biol.* **325**, 179–189.

Stern, C. D., Fraser, S. E., Keynes, R. J., and Primmett, D. R. (1988). A cell lineage analysis of segmentation in the chick embryo. *Development* **104S**, 231–244.

Takahashi, Y., Inoue, T., Gossler, A., and Saga, Y. (2003). Feedback loops comprising Dll1, Dll3 and Mesp2, and differential involvement of Psen1 are essential for rostrocaudal patterning of somites. *Development* **130**, 4259–4268.

Wood, A., and Thorogood, P. (1994). Patterns of cell behavior underlying somitogenesis and notochord formation in intact vertebrate embryos. *Dev. Dynam.* **201**, 151–167.

Xu, Q. L., Mellitzer, G., Robinson, V., and Wilkinson, D. G. (1999). In vivo cell sorting in complementary segmental domains mediated by Eph receptors and ephrins. *Nature* **399**, 267–271.

Xu, Q. L., Mellitzer, G., and Wilkinson, D. G. (2000). Roles of Eph receptors and ephrins in segmental patterning. *Philos. Trans. Roy. Soc. London B Biol. Sci.* **355**, 993–1002.

Zeng, W., Thomas, G. L., and Glazier, J. A. (2004). NonTuring stripes and spots: A novel mechanism for biological cell clustering. *Physica A* **341**, 482–494.