The Company of Biologists

RESEARCH ARTICLE

The *Pax6* master control gene initiates spontaneous retinal development via a self-organising Turing network

Timothy Grocott*, Estefania Lozano-Velasco, Gi Fay Mok and Andrea E. Münsterberg

ABSTRACT

Understanding how complex organ systems are assembled from simple embryonic tissues is a major challenge. Across the animal kingdom a great diversity of visual organs are initiated by a 'master control gene' called *Pax6*, which is both necessary and sufficient for eye development. Yet precisely how *Pax6* achieves this deeply homologous function is poorly understood. Using the chick as a model organism, we show that vertebrate *Pax6* interacts with a pair of morphogen-coding genes, *Tgfb2* and *Fst*, to form a putative Turing network, which we have computationally modelled. Computer simulations suggest that this gene network is sufficient to spontaneously polarise the developing retina, establishing the first organisational axis of the eye and prefiguring its further development. Our findings reveal how retinal self-organisation may be initiated independently of the highly ordered tissue interactions that help to assemble the eye *in vivo*. These results help to explain how stem cell aggregates spontaneously self-organise into functional eye-cups *in vitro*. We anticipate these findings will help to underpin retinal organoid technology, which holds much promise as a platform for disease modelling, drug development and regenerative therapies.

KEY WORDS: Eye development, Follistatin, Pattern formation, Pax6, Self-organisation, TGFβ

INTRODUCTION

Positional cues that govern cell fate decisions in the embryo may arise at multiple organisational levels: cell intrinsically (e.g. asymmetric cell divisions), tissue intrinsically (e.g. reaction-diffusion mechanisms), tissue extrinsically (e.g. inductive tissue interactions) or some combination of these. Historically, the early patterning of cell fates within the vertebrate eye has emphasised inductive interactions, stemming from Spemann's seminal work on lens induction (Spemann, 1901). These inductive interactions furnish positional information to coordinate self-assembly of the various tissues that make up the vertebrate camera eye, including the optic vesicle of the forebrain, which generates the retina, and the overlying presumptive lens ectoderm (Gunhaga, 2011). In the embryo, interactions with neighbouring tissues help to remodel the hemi-spherical optic vesicle into a bi-layered optic cup

School of Biological Sciences, University of East Anglia, Norwich Research Park, Norwich NR4 7TJ, UK.

*Author for correspondence (t.grocott@uea.ac.uk)

T.G., 0000-0002-6321-401X; E.L.-V., 0000-0002-5615-2754; G.F.M., 0000-0002-5202-9062; A.E.M., 0000-0002-4577-4240

(Fig. 1A). Yet this vesicle-to-cup transformation is spontaneously recapitulated by stem cell-derived retinal organoids *in vitro* (Eiraku et al., 2011), revealing that a hitherto unsuspected tissue-intrinsic mechanism suffices to self-organise the primary retinal axis. Here, we provide evidence for a self-organising mechanism centred on the transcription factor-coding gene paired box 6 (*Pax6*).

Pax6 has been called an eye master control gene (Gehring, 1996) and is necessary for eye development across much of the animal kingdom, from flies to humans (Hill et al., 1991; Hodgson and Saunders, 1980; Hoge, 1915; Nakayama et al., 2015). Mis-expression of mammalian or cephalopod *Pax6* genes triggers the spontaneous development of ectopic compound eyes in arthropods (Halder et al., 1995; Tomarev et al., 1997), as well as supernumerary camera eyes in vertebrates (Chow et al., 1999). This deeply homologous function, whereby a shared *Pax6* genetic apparatus builds eye structures that are morphologically and phylogenetically distinct (Shubin et al., 1997), is poorly understood.

The transforming growth factor β (Tgfβ) signalling pathway (Massagué, 1998) is transduced by ligand dimers that assemble hetero-tetrameric receptor complexes. The activated receptor complex then phosphorylates Smad2 and Smad3 proteins, which assemble with Smad4 before translocating to the nucleus where they interact with transcription factors to regulate gene expression. Whereas Smad2, Smad3 and Smad4 transduce Tgfβ, activin and Nodal signals, an inhibitory Smad7 antagonises this pathway cell-autonomously. Additionally, secreted antagonists, such as follistatin (Fst) act non-cell-autonomously by blocking ligand-receptor interactions (Iemura et al., 1998; Nakamura et al., 1990; Nogai et al., 2008). Smad4 is shared with the parallel bone morphogenetic protein (Bmp) signalling pathway, the signals of which are transduced by Smad1, Smad5 and Smad8, and inhibited by Smad6. We have previously reported that Pax6 protein function, and thus autoregulation, is inhibited via a direct Tgfβ-dependent interaction with Smad3, which inhibits Pax6-DNA binding (Grocott et al., 2007). Subsequently, we showed that Tgfβ signals emanating from the peri-ocular neural crest mesenchyme suppress Pax6 to align the lens with the optic vesicle (Grocott et al., 2011).

The molecular mechanisms by which tissues spontaneously generate patterns was first considered by Turing, who coined the term 'morphogen' to describe such molecules and devised reaction-diffusion models to simulate them (Turing, 1952). Gierer and Meinhardt later independently conceived of their activator-inhibitor model – a Turing network in which a slow-diffusing activator morphogen drives both its own production and that of a faster diffusing inhibitor morphogen, which suppresses the activator (Fig. 1B) (Gierer and Meinhardt, 1972). Thus, there arises a molar excess of activator over inhibitor at their source, where positive-feedback dominates, but a molar excess of inhibitor away from their source, where negative-feedback dominates (Fig. 1C).
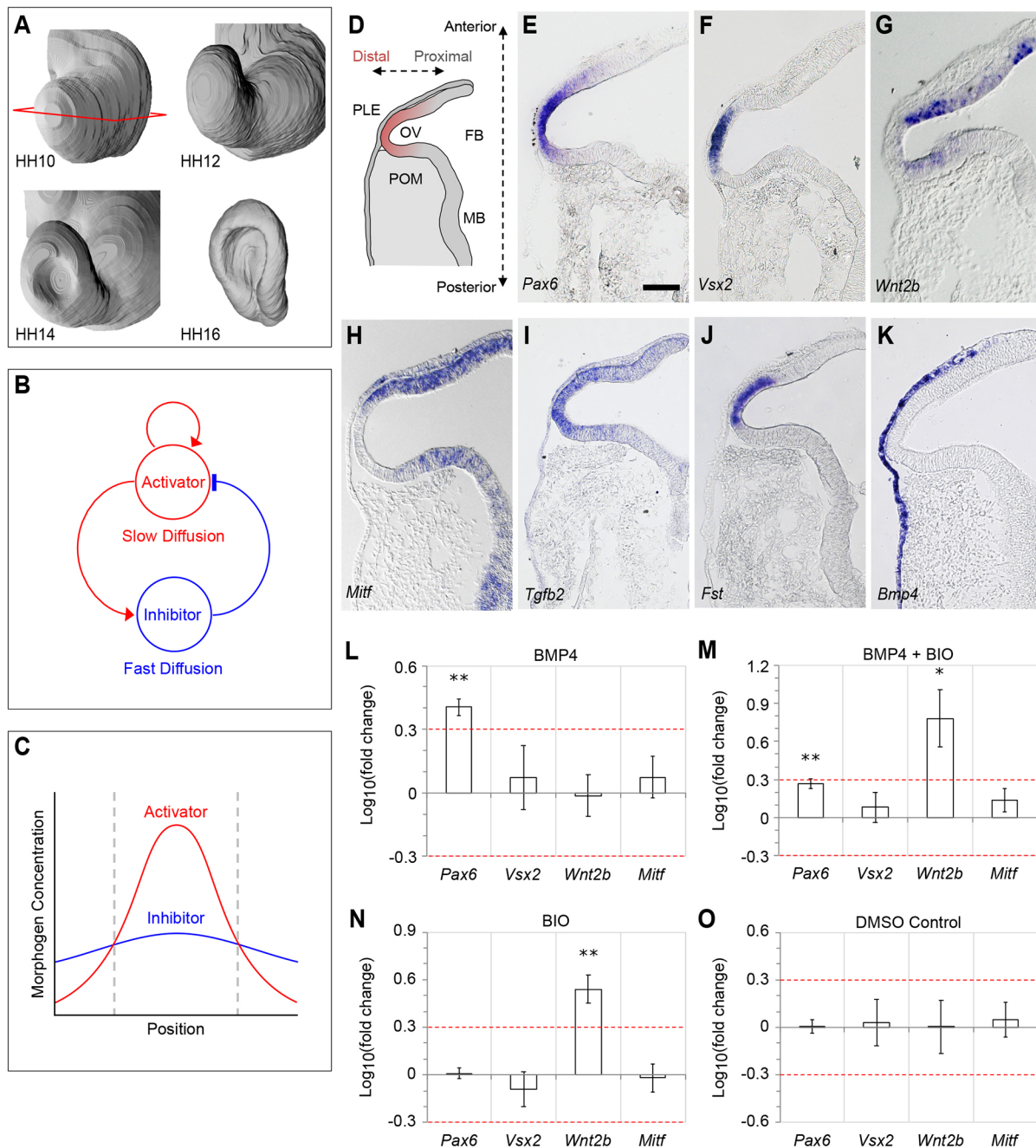
Fig. 1. Bmp and canonical Wnt signalling do not directly synergise to induce proximal identity in the optic vesicle. (A) 3D surface reconstructions of the chick optic vesicle/cup from stages HH10-HH16. The horizontal plane of sectioning is indicated for stage HH10. (B,C) An activator-inhibitor type Turing network. (B) A slow-diffusing activator morphogen drives its own production and that of a faster-diffusing inhibitor morphogen, which inhibits the activator. (C) The network yields a molar excess of activator over inhibitor at their common source, but an excess of inhibitor away from their source. (D) Schematic representation of a horizontal section through the stage HH10 chick optic vesicle identifying neighbouring tissues, anterior-posterior axis and proximal-distal axis. OV, optic vesicle; PLE, presumptive lens ectoderm; POM, periocular mesenchyme; FB, forebrain; MB, midbrain. (E-K) The HH10 optic vesicle is polarised along a proximal-distal axis. Horizontal sections reveal polarised expression of the marker genes (E) *Pax6*, (F) *Vsx2*, (G) *Wnt2b*, (H) *Mitf*, (I) *Tgfb2* and (J) *Fst*. (K) *Bmp4* is expressed in the overlying presumptive lens ectoderm. (L-O) RT-QPCR analysis of proximal and distal marker gene expression following 16 h exposure to (L) Bmp4 only, (M) Bmp4 and BIO (a canonical Wnt agonist) in combination, (N) BIO only and (O) DMSO carrier control. Values plotted are Log10(mean fold change)±s.e.m. Red guidelines indicate the levels of ±2-fold change in gene expression. *$P<0.05$, **$P<0.01$ (Student's paired t-test). Scale bars: 150 μm.

Here, we describe a putative self-organising Turing network (Turing, 1952) comprising *Pax6* and a pair of morphogen-coding genes: transforming growth factor β2 (*Tgfb2*) and follistatin (*Fst*). Using reaction-diffusion modelling we show how this gene network may spontaneously polarise the optic vesicle to trigger self-organisation of the vertebrate retina.

RESULTS

Extrinsic Bmp signals drive *Pax6* expression in the distal optic vesicle

Optic vesicle polarisation is apparent from Hamburger and Hamilton (1992) stage HH10 in the chick, evidenced by differential gene expression along a proximal-distal axis (Fig. 1D): *Pax6* and visual

2

system homeobox 2 (*Vsx2*; formerly *Chx10*) are expressed distally (Fig. 1E,F), whereas microphthalmia-associated transcription factor (*Mitf*) and Wnt family member 2b (*Wnt2b*; formerly *Wnt13*) are expressed proximally (Fig. 1G,H). We additionally report that two further genes, transforming growth factor β2 (*Tgfb2*) and follistatin (*Fst*) are co-expressed with *Pax6* in the distal optic vesicle (Fig. 1I,J). Neither *Tgfb2* nor *Fst* expression is detected in the overlying presumptive lens ectoderm.

As the optic vesicles evaginate between stages HH8 and HH10, they encounter bone morphogenetic protein (Bmp) family growth factors from the overlying surface ectoderm (e.g. *Bmp4*; Fig. 1K). Bmps are implicated in establishing both distal and proximal cell identities within the optic vesicle; Bmp alone promotes distal character (Pandit et al., 2015), whereas combined with canonical Wnt signalling it was proposed to induce proximal character (Steinfeld et al., 2013). Consistently, we found that exposing HH10 optic vesicle explants to Bmp4 ligand for 16 h *in vitro* led to an upregulation of distal *Pax6* (2.35±0.19 fold, mean±standard deviation; $P<0.01$; $n=4$) as measured by RT-QPCR (Fig. 1L). The remaining distal (*Vsx2*) and proximal (*Wnt2b* and *Mitf*) markers were not significantly affected (Fig. 1L). Following combined exposure to both Bmp4 and the Wnt agonist BIO (6-bromoindirubin-3′-oxime; GSK3 inhibitor) (Meijer et al., 2003), *Pax6* (1.88±0.38 fold; $P<0.05$; $n=5$) was similarly affected (Fig. 1M), while the proximal marker *Wnt2b* was additionally upregulated (9.28±7.89 fold; $P<0.05$; $n=5$), suggesting

that *Wnt2b* may auto-regulate. Wnt activation alone induced proximal *Wnt2b* (3.69±1.43 fold; $P<0.01$; $n=4$) without significantly affecting distal markers (Fig. 1N), while exposure to DMSO (carrier for BIO) had no impact (Fig. 1O). These data do not support a direct synergism between Bmp and Wnt signalling in establishing proximal-distal polarity, as their combined action is merely additive.

To validate the interaction between Bmp signalling and *Pax6* expression *in vivo*, we performed electroporation-mediated gene transfer to mis-express the cell-autonomous Bmp inhibitor *Smad6* in single optic vesicles, while unelectroporated contralateral vesicles served as internal negative controls (Fig. 2A). In comparison with mis-expression of a benign enhanced green fluorescent protein (GFP; 1.13±0.37 fold; $n=7$; Fig. 2C,D), *Smad6* caused an asymmetric reduction in the area of *Pax6* expression between transfected and contralateral control vesicles (0.56±0.31 fold; $P<0.05$; $n=13$; Fig. 2C,E). This confirms that distal *Pax6* expression *in vivo* requires upstream Bmp.

Auto-regulation of *Pax6* has been reported in a number of tissues, including the lens (Ashery-Padan et al., 2000). To test for *Pax6* auto-regulation in the optic vesicle, a C-terminally truncated dominant-negative *Pax6* gene (*dnPax6*) (Grocott et al., 2007) was mis-expressed unilaterally, while a C-terminal riboprobe was used to selectively detect endogenous *Pax6* expression (Fig. 2B). *dnPax6* did not disrupt endogenous *Pax6* expression (0.75±0.36 fold; $P>0.05$; $n=9$; Fig. 2C,F) compared with the GFP control, nor could
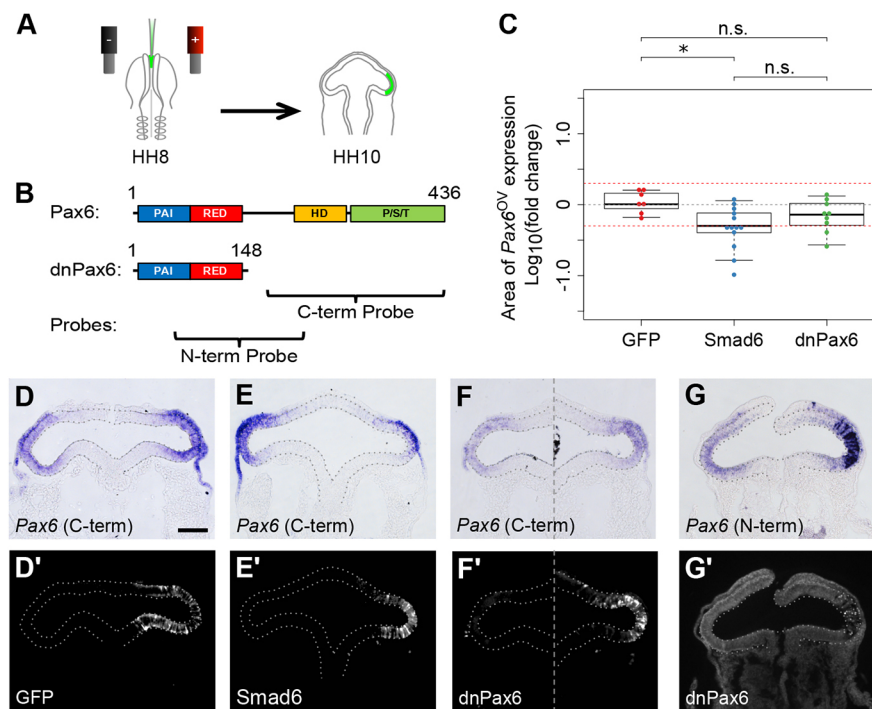

Fig. 2. Bmp signalling is required for *Pax6* gene expression in the distal optic vesicle. (A) DNA expression constructs were injected into the lumen of the anterior neural tube of stage HH8 chick embryos and electroporated to transfect a single prospective optic vesicle, the other serving as an untransfected internal control. Embryos were cultured for 10-12 h overnight until stage HH10 when they were analysed. (B) Schematic showing the domain structure of the major Pax6 isoform compared with the truncated dominant-negative Pax6 (dnPax6). PAI and RED, DNA-binding subdomains that make up the N-terminal paired domain; HD, DNA-binding homeodomain; P/S/T, C-terminal proline/serine/threonine-rich transactivation domain. Antisense RNA probes against C- or N-terminal sequences respectively detect endogenous Pax6 transcripts only or endogenous Pax6 and dnPax6 together. (C) The sectional area of *Pax6* gene expression was measured and compared between electroporated and non-electroporated optic vesicles for each embryo. Log10(fold change) was plotted for embryos electroporated with GFP control construct, Smad6+GFP construct or dnPax6+GFP construct. Red guidelines indicate the level of ±2-fold change in sectional expression area. *$P<0.05$; n.s. indicates $P≥0.05$ (one-way ANOVA with Tukey's post-hoc test). (D-G) Endogenous *Pax6* gene expression following transfection with (D) GFP control, (E) Smad6+GFP and (F,G) dnPax6+GFP, and (D′-G′) anti-GFP immunofluorescence showing the location of (D′) GFP transfected cells, (E′) Smad6+GFP transfected cells and (F′,G′) dnPax6+GFP transfected cells. Scale bars: 100 μm. Immunofluorescence in G′ is heavily quenched by strong *in situ* staining. Optic vesicles are indicated by broken outlines.

we distinguish a difference between *dnPax6* and *Smad6* mis-expression (Fig. 2C; *P*>0.05). To confirm that *dnPax6* was overexpressed relative to endogenous *Pax6*, an N-terminal riboprobe was used to collectively detect both endogenous *Pax6* and exogenous *dnPax6* expression (Fig. 2G). Thus, although distal *Pax6* expression in the optic vesicle requires Bmp signalling *in vivo*, we cannot exclude the possibility that upstream Bmp action may mask subsequent *Pax6* auto-regulation.

Pax6 drives expression of *Tgfb2* and its antagonist *Fst* in the distal optic vesicle

Migratory neural crest cells reach the optic vesicle at stage HH10, where they contribute to the periocular mesenchyme and are thought to induce proximal and suppress distal gene expression via Tgfβ subfamily signalling (Fuhrmann et al., 2000; Grocott et al., 2011). Exogenously supplied Tgfβ subfamily ligand (activin A) was reported to induce proximal (*Wnt2b* and *Mitf*) and inhibit distal (*Pax6* and *Vsx2*) gene expression in explant cultures (Fuhrmann et al., 2000). In contrast to this tissue-extrinsic induction mechanism, stem cell-derived retinal organoids are reported to polarise tissue-autonomously, exemplified by the spontaneous acquisition of proximal Wnt activity (Hasegawa et al., 2016). This raises the possibility of a redundant tissue-intrinsic polarising activity. Given that distal *Tgfb2* expression correlates with *Pax6* (Fig. 1E,I) we asked whether *Pax6* might induce *Tgfb2* to activate proximal target genes tissue-autonomously. In comparison with GFP controls (1.06±0.17 fold; *n*=8; Fig. 3A,B), mis-expression of *dnPax6* in single optic vesicles diminished *Tgfb2* expression relative to contralateral control vesicles (0.79±0.54 fold; *P*<0.05; *n*=15; Fig. 3A,C). Thus, the *Pax6* master controller is required for *Tgfb2* expression in the distal vesicle, consistent with a report of Pax6-binding sites located within the *Tgfb2* promoter (Wolf et al., 2009).

This presents a paradox, however: *Tgfb2* expression (Fig. 1I) negatively correlates with its positive targets *Wnt2b* and *Mitf* (Fig. 1G,H), yet positively correlates with its negative targets *Pax6* and *Vsx2* (Fig. 1E,F) (Fuhrmann et al., 2000). How might Tgfβ pathway activation become inverted relative to *Tgfb2* gene

expression? We considered whether *Pax6* might also activate *Fst* (Fig. 1J), a Tgfβ antagonist, to grant distal immunity from Tgfβ signalling. Compared with GFP controls (1.31±0.63 fold; *n*=6; Fig. 3D,E), mis-expression of *dnPax6* in a single optic vesicle significantly reduced *Fst* expression (0.69±0.34 fold; *P*<0.05; *n*=8; Fig. 3D,F). Thus, *Pax6* function is additionally required for *Fst* expression in the distal vesicle.

The paradoxical out-of-phase expression of distal *Tgfb2* and its proximal (positive) targets might then be explained by differential diffusion of *Tgfb2* and *Fst* gene products resulting in: (1) Tgfβ2 being locally sequestered by slow-diffusing Fst within the distal vesicle, thereby preserving distal character; and (2) fast-diffusing Tgfb2 dispersing proximally away from Fst, to induce proximal character within the neighbouring proximal vesicle.

To test whether this hypothesis is plausible, we examined a reaction-diffusion model of the interactions summarised in Fig. 4A (Model A, Fig. S1; see supplementary information) and performed numerical simulations in one dimension only to represent the anterior-posterior axis (comprising anterior-proximal, distal and posterior-proximal domains) of the optic vesicle. Simulations were performed with both zero-flux (Fig. 5) and periodic (Movies 1 and 2) boundary conditions to represent dissected optic vesicle explants and spherical organoids, respectively.

A variety of diffusion ratios for Tgfb2 dimers and Fst monomers versus Fst:Tgfb2 complexes were explored (e.g. Fig. 4B,B′; Fig. S2; Movie 1). Simulations demonstrated that local inhibition and lateral-activation of Tgfb signalling may occur if the diffusion rate of Fst: Tgfb2 complexes exceeds that of Fst monomers. Although initially counter-intuitive, there is precedent for ligand:antagonist complexes that disperse faster than their individual constituents (Esteve et al., 2011), and our subsequent simulations assume this condition is satisfied.

Pax6, *Fst* and *Tgfb2* form a self-organising Turing network that can dynamically polarise the optic vesicle

Given that Tgfb signalling is known to disrupt Pax6 protein function (Grocott et al., 2007), such local inhibition and lateral activation of
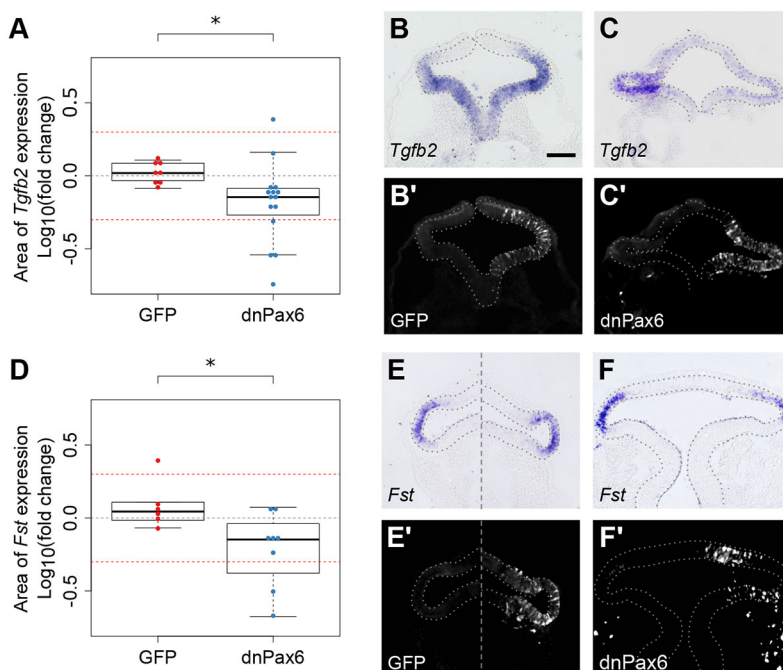
Fig. 3. Pax6 function is required for expression of *Tgfb2* and *Fst*. (A-C′) *Tgfb2* gene expression was assessed 12 h after electroporation of GFP or dnPax6+GFP into a single optic vesicle. (A) Sectional areas of *Tgfb2* gene expression were measured and compared between electroporated and non-electroporated optic vesicles for each embryo. Log10(fold change) was plotted for each embryo. Red guidelines indicate the level of ±2-fold change in sectional expression area. (B,C) *Tgfb2* gene expression following electroporation with (B) GFP control and (C) dnPax6+GFP, and (B′,C′) anti-GFP immunofluorescence showing the location of (B′) GFP transfected cells and (C′) dnPax6+GFP transfected cells. (D-F) *Fst* expression was assessed 12 h after electroporation with GFP or dnPax6+GFP. (D) Sectional areas of *Fst* gene expression were measured and compared between electroporated and non-electroporated optic vesicles for each embryo. Log10(fold change) was plotted for each embryo. Red guidelines indicate the level of ±2-fold change in sectional expression area. (E,F) *Fst* gene expression following electroporation with (E) GFP control and (F) dnPax6+GFP, and (E′,F′) anti-GFP immunostaining showing the location of (E′) GFP transfected cells and (F′) dnPax6+GFP transfected cells. Optic vesicles are indicated by broken outlines. *P<0.05 (Welch's two-sample *t*-test). Scale bars: 100 μm.

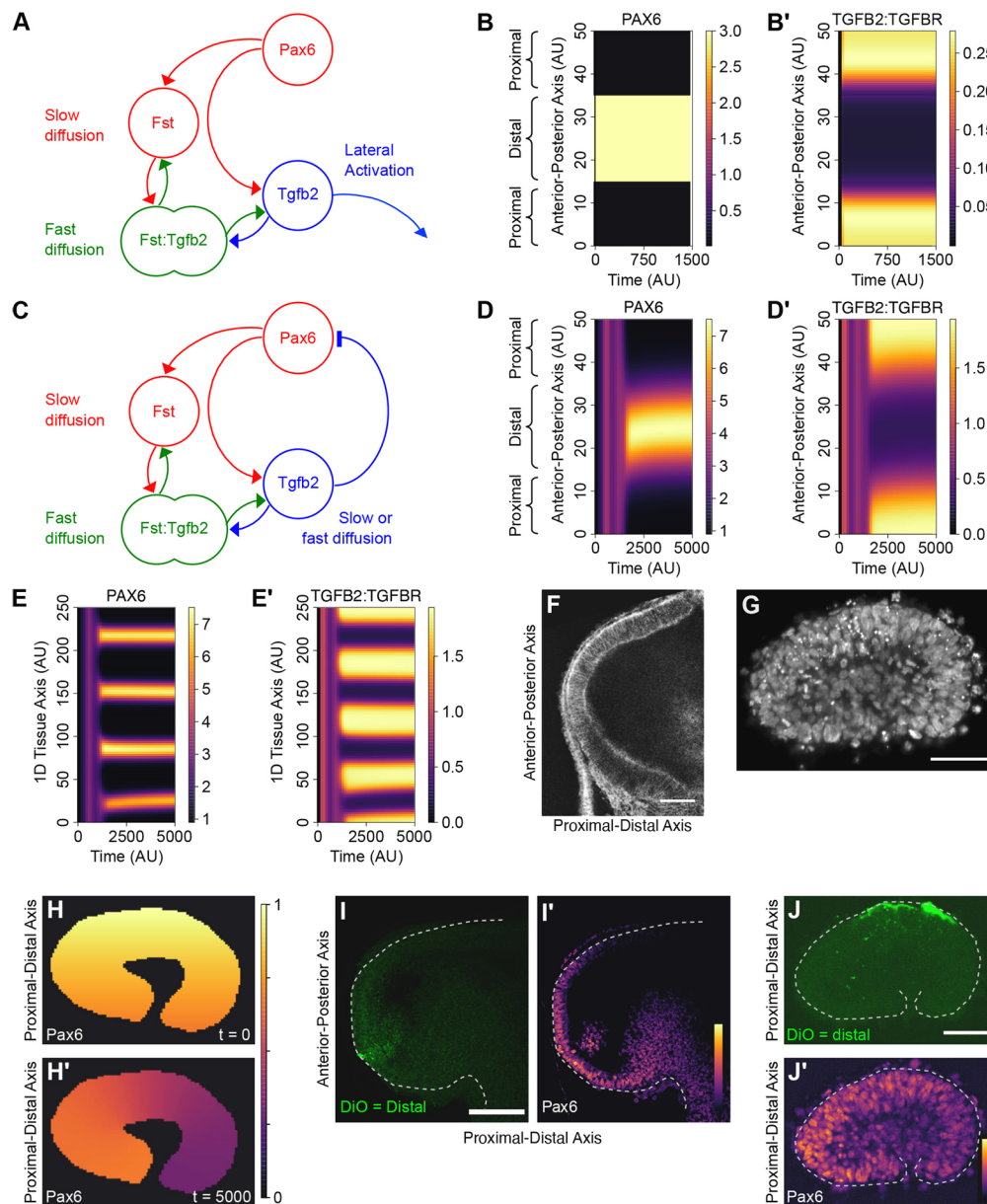**Fig. 4. Reaction-diffusion modelling of the *Pax6*/*Fst*/*Tgfb2* gene network.** (A) Summary of Model A, in which Pax6 drives expression of both *Fst* and *Tgfb2*, whereas Fst inhibits Tgfβ2 function via sequestration. Slow diffusion of Fst was postulated to result in local inhibition of Tgfβ2 at the source of *Pax6*/*Tgfb2*/*Fst* expression. Conversely, fast diffusion of Tgfβ2 was postulated to drive lateral activation of its downstream signalling pathway away from the *Pax6*/*Fst*/*Tgfb2*-expressing region. (B,B′) 1D numerical simulation of Model A, in which Pax6 expression is regionally restricted throughout. For all simulations, units of space, time and molecular concentrations are arbitrary. The vertical *y*-axis represents the anterior-posterior axis of the hemispherical optic vesicle, which is divided into anterior-proximal, distal and posterior-proximal domains. The plots depict the time evolution (*x*-axis) for 1D spatial distributions (*y*-axis) of (B) Pax6 and (B′) the activated Tgfβ2:Tgfβ-receptor signalling complex. (C) Summary of Model B in which Fst:Tgfβ2 complex quickly diffuses and dissociates, while Tgfβ2 additionally inhibits Pax6 transcriptional activator function. (D,D′) 1D numerical simulation of Model B in which Pax6 expression is initially homogenous but noisy. The plots depict spontaneous generation of (D) a Pax6+ 'distal pole' flanked by (D′) Tgfβ2:Tgfβ-receptor+ 'proximal poles'. (E,E′) 1D numerical simulation of Model B with a larger tissue size resulting in (E) multiple Pax6+ 'distal poles' interspersed with (E′) Tgfβ2:Tgfβ-receptor+ 'distal poles'. (F) Confocal section of an HH10 tg(membrane-GFP) embryo showing optic vesicle size prior to explant culture. (G) Confocal section of a fixed optic vesicle explant showing the collapsed tissue following 16 h culture. Cell nuclei are stained with DAPI. (H,H′) 2D numerical simulation of Model B within an explant-shaped domain (Model C). (H) The initial distal-high to proximal-low Pax6 pattern is (H′) dynamically re-polarised along the longest axis of the explant. (I,I′) Partially dissected optic vesicle in which the distal end was fluorescently labelled with (I) DiO, corresponding to (I′) the Pax6+ pole revealed by immunofluorescent staining. (J,J′) Explant experiment in which (J) the distal pole of the optic vesicle was labelled with DiO during dissection. (J′) Following overnight culture, Pax6 expression has re-polarised relative to the former proximal-distal axis. Scale bars: 50 µm in F,G,J; 100 µm in I.

Tgfβ signalling equate to local positive feedback and lateral-inhibition of the *Pax6* master control gene, respectively (Fig. 4C). This is functionally equivalent to a simple activator-inhibitor type (Fig. 1B) Turing network (Gierer and Meinhardt, 1972; Turing, 1952), which

can serve as a spontaneous pattern generator: *Pax6* and *Fst* provide a short-range auto-regulating activator; and *Tgfb2* is the long-range inhibitor (compare Fig. 1B with Fig. 4C). To explore whether the network of Fig. 4C possesses spontaneous polarising activity,
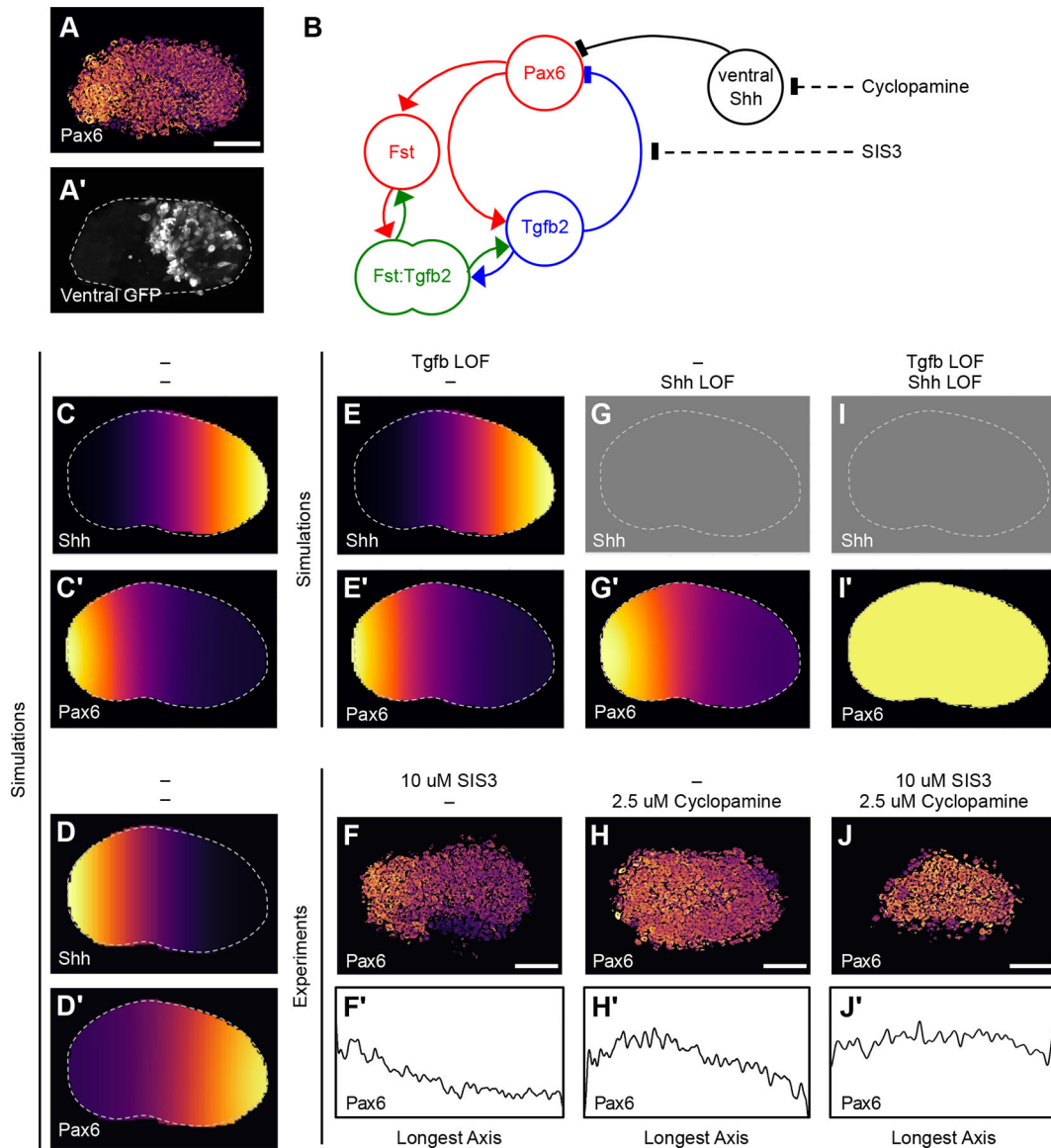
Fig. 5. Shh positional information and Tgfβ-mediated self-organisation position the Pax6+ pole in cultured explants. (A,A′) The Pax6+ pole re-aligns with the dorsal-ventral axis in explanted optic vesicles. Maximal projections of (A) Pax6 immunofluorescence normalised to DAPI and (A′) ventrally targeted GFP in a whole-mount explant. (B) Summary of Model D in which a ventral-high to dorsal-low Shh gradient inhibits Pax6 expression. The pharmacological compounds used in functional experiments are also indicated (broken lines). (C,C′) 2D numerical simulation of Model D showing (C) the ventral-high Shh gradient (C′) Pax6 re-polarisation. (D,D′) 2D numerical simulation of Model D showing (D) reversal of the Shh gradient and (D′) corresponding reversal of Pax6 polarity. (E,E′) 2D numerical simulation of Model D with Tgfβ loss of function (Tgfb LOF) showing (E) the ventral high Shh gradient and (E′) the resulting Pax6 distribution. (F,F′) Optic vesicle explants were cultured with 10 μM SIS3 for 16 h. (F) Maximum projection of Pax6 immunofluorescence normalised to DAPI. (F′) 1D profile plot of Pax6 abundance along the longest (horizontal) axis of the explant. (G,G′) 2D numerical simulation of Model D with Shh loss of function (Shh LOF) showing (G) absence of Shh positional information and (G′) the resulting Pax6 distribution. (H,H′) Optic vesicle explants were cultured with 2.5 μM cyclopamine for 16 h. (H) Maximum projection of Pax6 immunofluorescence normalised to DAPI. (H′) 1D profile plot of Pax6 abundance along the longest (horizontal) axis of the explant. (I,I′) 2D numerical simulation of Model D with both Tgfβ loss of function and Shh loss of function showing (I) absence of Shh positional information and (I′) the resulting Pax6 distribution. (J,J′) Optic vesicle explants were cultured with both 10 μM SIS3 and 2.5 μM cyclopamine for 16 h. (J) Maximum projection of Pax6 immunofluorescence normalised to DAPI. (J′) 1D profile plot of Pax6 abundance along the longest (horizontal) axis of the explant. Scale bars: 50 μm.

we simply extended Model A to include inhibition of Pax6 function by Tgfβ signalling (Model B; Fig. S3; see supplementary information). Simulations showed that an initially homogenous but noisy *Pax6* distribution is readily converted into a polarised pattern, wherein *Pax6* expression becomes regionally restricted (Fig. 4D) and out-of-phase with Tgfb receptor activation (Fig. 4D′; Fig. S4; Movie 2). Additionally, simulating larger tissue sizes results not in a larger *Pax6*-expressing distal pole, but in a greater number of *Pax6*-expressing distal poles of approximately equal size (Fig. 4E,E′). This hallmark feature of Turing networks is remarkably consistent with observations of retinal organoid cultures in which stem cell aggregates yielded between one and four retinas each (Eiraku et al., 2011).

Similarly, reducing tissue size limits the number rather than the size of pattern elements generated by a Turing network so that, for example, a single 'spot', half a 'spot' (i.e. a gradient) or no 'spot' is generated. When cultured as isolated explants in the absence of

serum, polarised HH10 optic vesicles (e.g. Fig. 4F) collapse into compact spheroids (Fig. 4G), reducing the longest dimension of this tissue to ≤0.5 fold. To better understand how the *Pax6*/*Fst*/*Tgfb2* network might respond in this situation, we performed 2D simulations of Model B on an explant-shaped domain with an initial distal-high to proximal-low Pax6 pattern (Model C; Fig. 4H; see supplementary information). For these simulations, we explored both zero-flux and fixed-boundary conditions, disregarding the latter as the former agreed more closely with experimental observations. It may be interpreted that adsorption of morphogens to extracellular matrix and cell-surface proteins within explants prevents a significant outward flux, while the absence of morphogens from the defined bathing medium prevents an inward flux.

Owing to the reduced tissue size, this proximal-distal pattern proved unstable and Pax6 expression quickly re-polarised to form a gradient along the longest axis of the explant (i.e. perpendicular to the former proximal-distal axis; Fig. 4H′; Fig. S5; Fig. S6; Movie 3). To test this model prediction, optic vesicles were dissected for explant culture, during which their distal poles were labelled with DiO (Fig. 4I). Immunostaining of a partially dissected optic vesicle verifies that DiO labelling coincides with the initial Pax6+ distal pole (Fig. 4I′). Following overnight culture, however, Pax6 expression no longer coincides with the distal DiO label but instead re-polarises along the longest axis of each explant (Fig. 4J,J′) consistent with simulations. This suggests that the *Pax6*/*Fst*/*Tgfb2* network can dynamically repolarise its expression in a self-organising fashion.

Intrinsic positional information constrains *Pax6*/*Fst*/*Tgfb2* self-organisation

In explant culture, optic vesicles are isolated from inductive tissue interactions and thus from extrinsic positional information. However, we questioned whether Pax6 repolarisation might be influenced by intrinsic positional information. A ventral-high to dorsal-low gradient of Sonic hedgehog (Shh) signalling activity exists within the optic vesicle, which is known to restrict the ventral extent of *Pax6* expression (Ekker et al., 1995; Macdonald et al., 1995). Might Shh positional information push the Pax6+ pole towards the dorsal side of the explant? Electroporation of a GFP expression construct was targeted to the ventral optic vesicle prior to dissection and overnight explant culture. Whole-mount immunofluorescence staining showed that the Pax6+ pole negatively correlates with ventral GFP expression (100% of explants; *n*=5; Fig. 5A,A′), supporting this idea.

To explore how Shh positional information might interact with the *Pax6*/*Fst*/*Tgfb* network, we extended Model C by incorporating Shh suppression of *Pax6* into the governing equations (Model D; Fig. 5B; see supplementary information) while adding a Shh positional information gradient (Fig. 5C; Fig. S7). Simulations showed that the Pax6+ pole reorientates away from the ventral-high end of the Shh gradient (Fig. 5C′; Fig. S8; Fig. S9) as was observed experimentally (Fig. 5A,A′). Moreover, inverting the Shh gradient (Fig. 5D) caused a reversal of *Pax6* polarity (Fig. 5D′).

Exploring Model D, we next simulated the ability of *Pax6* to repolarise in the absence of Tgfβ-mediated self-organisation and found that the Shh positional information gradient was sufficient to generate a dorsal Pax6+ pole (Fig. 5E,E′). This prediction was tested experimentally by culturing optic vesicle explants in the presence of a Smad3 inhibitor, SIS3 (Jinnin et al., 2006). As Tgfβ inhibits Pax6 protein function via its specific and direct interaction with Smad3 (Grocott et al., 2007), SIS3 should block the inhibition of Pax6 by Tgfβ2 (Fig. 5B). Following overnight culture with 10 μM SIS3, optic vesicle explants still exhibited distinct Pax6+ poles (91% of

explants; *n*=11; Fig. 5F,F′) as predicted. These data show that Tgfβ-mediated self-organisation is not required for *Pax6* polarisation in cultured explants, presumably due to the redundant action of Shh positional information.

Model D simulations lacking Shh positional information (Shh LOF; Fig. 5G) predicted that the *Pax6*/*Fst*/*Tgfb2* network should suffice to generate a Pax6+ pole in the absence of Shh activity (Fig. 5G′). To test this, optic vesicle explants were cultured overnight with 2.5 μM cyclopamine: a steroidal alkaloid that inhibits the Hedgehog pathway transducer Smoothened (Chen et al., 2002). As predicted, explants still exhibited Pax6+ poles in the absence of Shh activity (82% of explants; *n*=11; Fig. 5H,H′). Thus, Shh positional information is not required for *Pax6* polarisation in optic vesicle explants, suggesting that the *Pax6*/*Fst*/*Tgfb2* network is sufficient to self-organise the Pax6+ pole. However, although still polarised in the absence of Shh positional information, *Pax6* expression is subtly upregulated both in simulations (compare Fig. 5E′ with 5G′) and in experiments (compare Fig. 5F with 5H).

Further Model D simulations predicted that simultaneous loss of both Shh positional information (Fig. 5I) and Tgfβ-mediated self-organisation should prevent *Pax6* polarisation in cultured explants (Fig. 5I′); instead of polarising, Pax6 was expressed uniformly throughout the simulated explant. Consistent with this, optic vesicle explants cultured with both 2.5 μM cyclopamine and 10 μM SIS3 mostly failed to exhibit *Pax6* polarisation as expression was approximately uniform across their lengths (67% of explants; *n*=12; Fig. 5J,J′). In other words, the *Pax6*/*Fst*/*Tgfb2* network appears to be both sufficient (Fig. 5H,H′) and necessary (Fig. 5J,J′) to self-organise *Pax6* polarisation in the absence of positional information.

The *Pax6*/*Fst*/*Tgfb2* gene network regulates distal neural retinal identity *in vivo*

The preceding data suggest that, although the *Pax6*/*Fst*/*Tgfb2* network may freely self-organise in isolation (e.g. in retinal organoids), *in vivo* this network is constrained by intrinsic (e.g. Shh) and extrinsic (e.g. Bmp4) positional information to ensure correct alignment of the distal Pax6+ pole within the camera eye. Thus, functional perturbations *in vivo* are not expected to drive the kind of dynamic re-polarisation observed in cultured explants. How then might functional perturbation of the *Pax6*/*Fst*/*Tgfb2* network impact optic vesicle patterning *in vivo*?

According to our model, interference with *Fst* gene expression should de-repress Tgfβ signalling and inhibit Pax6 protein function in the distal vesicle, via the direct Tgfβ-dependent interaction of Smad3 with Pax6 (Grocott et al., 2007). Moreover, if Pax6 auto-regulates in the distal vesicle, this should manifest as a Tgfβ-mediated reduction in *Pax6* gene expression. To test this prediction, we employed morpholino oligonucleotides to suppress translation of Fst 315 and Fst 300 isoforms (Fig. 6A) within single optic vesicles. *Pax6* expression was then compared between these and unperturbed contralateral vesicles. Fst morpholino (FstMO) was first shown to suppress endogenous translation of both Fst isoforms in cultured chick embryonic cells via western blotting when compared with a standard control morpholino (StdMO) that does not target *Fst* (Fig. 6B).

In vivo, StdMO controls had no impact on *Pax6* expression in transfected optic vesicles (1.05±0.31 fold; *n*=20; Fig. 6C,D). In comparison, FstMO reduced *Pax6* expression in transfected vesicles (0.76±0.50 fold; *P*<0.01; *n*=18; Fig. 6C,E). We were able to rescue this loss of *Pax6* expression by co-transfecting FstMO together with an exogenous *Fst* transgene that evades FstMO and encodes the Fst
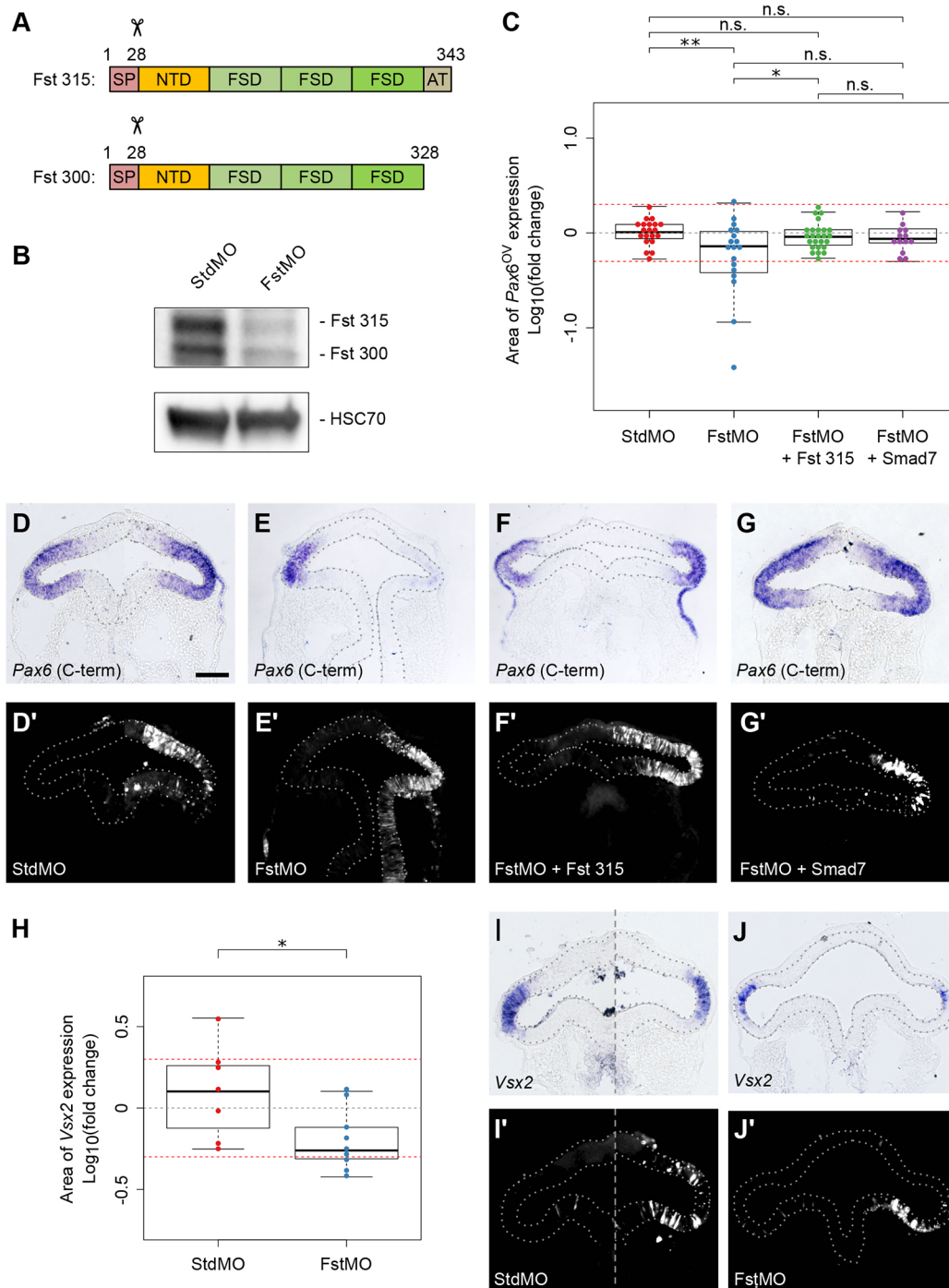
Fig. 6. *Fst* gene function is required for correct optic vesicle polarisation via distal inhibition of Tgfβ signalling. (A) Schematic showing domain structures encoded by naturally occurring Fst transcripts. The shorter Fst 300 is generated by alternative splicing. SP, 28 amino acid signal peptide cleaved co-translationally; NTD, N-terminal domain; FSD, follistatin domain; AT, acidic tail. (B) Western blot validation of Fst 315 and Fst 300 protein knockdown by FstMO but not by StdMO in cultured chick embryo cells. (C-G′) Sectional area of *Pax6* gene expression was assessed 12 h after co-electroporation of single optic vesicles with control/experimental morpholinos plus various gene expression constructs. (C) Sectional area of *Pax6* gene expression was measured and compared between electroporated and non-electroporated optic vesicles for each embryo. Log10(fold change) was plotted for each embryo. Red guidelines indicate the level of ±2-fold change in sectional expression area. (D-G) *Pax6* gene expression following co-electroporation of (D) standard control morpholino (StdMO)+GFP, (E) *Fst* morpholino (FstMO)+GFP, (F) FstMO+Fst gene expression construct and (G) FstMO+Smad7 gene expression construct. (D′-G′) FITC-labelled (D′) StdMO fluorescence, (E′) FstMO fluorescence, (F′) FstMO fluorescence and (G′), FstMO fluorescence, showing the location of transfected cells. (H-J′) Sectional area of *Vsx2* gene expression was assessed 12 h after co-electroporation of single optic vesicles with control/experimental morpholino. (H) Sectional area of *Vsx2* gene expression was measured and compared between electroporated and non-electroporated optic vesicles for each embryo. Log10(fold change) was plotted for each embryo. Red guidelines indicate the level of ±2-fold change in sectional expression area. (I,J) *Vsx2* gene expression following co-electroporation of (I) StdMO+GFP and (J) FstMO+GFP, and (I′,J′) FITC-labelled (I′) StdMO fluorescence and (J′) FstMO fluorescence showing the location of transfected cells. Optic vesicles are indicated by broken outlines. *$P<0.05$; **$P<0.01$ (C, one-way ANOVA with Tukey's post-hoc test; H, Welch's two-sample *t*-test). Scale bars: 100 μm.

315 isoform (0.98±0.35 fold; *P*>0.05; *n*=25; Fig. 6C,F). This confirmed that loss of *Pax6* was not due to a morpholino off-target effect and that *Fst* gene function is required for distal *Pax6* expression in the optic vesicle. This is consistent with earlier reports that neural induction by way of *Fst* overexpression induces *Pax6* in *Xenopus* animal cap explants (Altmann et al., 1997).

To verify that loss of *Pax6* expression is indeed due to the predicted de-repression of Tgfβ signalling, we attempted an alternate rescue by co-transfecting FstMO together with a cell-autonomous Tgfβ/Activin/Nodal pathway inhibitor, *Smad7*. As can be seen (Fig. 6C,G), no significant loss of *Pax6* expression was observed (0.91±0.31 fold; *P*>0.05; *n*=13) when Fst translation and Tgfb signalling were simultaneously suppressed.

In addition to inducing *Pax6* (Altmann et al., 1997), overexpression of *Fst* in *Xenopus* animal cap explants was reported to induce expression of the retinal photoreceptor marker *Opsin* (Hemmati-Brivanlou et al., 1994). We therefore investigated whether *Vsx2*, a distally expressed neural retinal marker (Liu et al., 1994) (Fig. 1F), is similarly affected upon disruption of the *Pax6/Fst/Tgfb2* gene network. In comparison with StdMO controls (1.51±1.05 fold; *n*=7; Fig. 6H,I), FstMO significantly reduced distal *Vsx2* expression in transfected optic vesicles (0.69±0.33 fold; *P*<0.05; *n*=9; Fig. 6H,J). Thus, de-repression of endogenous Tgfβ signalling in the distal vesicle is detrimental for correct proximal-distal patterning, including specification of the neural retina. These results are consistent with our general model and support the idea that Fst and Tgfb2 morphogens positively and negatively regulate Pax6 function, respectively, in order to polarise the optic vesicle.

DISCUSSION
The issue of the master control mechanism of *Pax6* has now been unresolved for a quarter of a century (Cvekl and Callaerts, 2017). Here, we have shown that the vertebrate *Pax6* directs expression of a pair of morphogen coding genes, *Fst* and *Tgfb2*, which modulate Pax6 function via positive and negative feedback. This *Pax6/Fst/Tgfb2* gene network topology is consistent with an activator-inhibitor type Turing network and appears to exhibit a self-organising pattern-forming ability in the absence of positional information. This spontaneous pattern-forming potential could explain both the ability of *Pax6* to trigger ectopic eye development across the animal kingdom (Chow et al., 1999; Halder et al., 1995;

Tomarev et al., 1997) and the spontaneous development of self-organising optic cups from stem cell aggregates cultured *in vitro* (Eiraku et al., 2011).

Prerequisites for retinal self-organisation
Our reaction-diffusion simulations showed that the *Pax6/Fst/Tgfb2* gene network may act as a self-organising Turing network, providing certain assumptions are satisfied. For example, we have assumed that larger Fst:Tgfb2 complexes diffuse more quickly than smaller Fst monomers. This is counter-intuitive as pure diffusion rate is a function of molecular mass. Yet there is a precedent for this phenomenon: for example, Sfrp:Wnt complexes have been observed to diffuse further than Wnt alone (Esteve et al., 2011). We postulate that Fst monomers disperse sub-diffusively due to binding interactions with extracellular matrix components and/or cell surface factors, e.g. heparin sulfate proteoglycans (Nakamura et al., 1991) or fibronectin (Maguer-Satta et al., 2006). In the context of Fst:Tgfb2 complexes, the relevant interaction surfaces may be shielded, enabling the larger complex to disperse further and faster than its constituents.

This assumed rapid dispersal of Fst:Tgfb2 complexes is only required if Tgfb2 sequestration by Fst is reversible, which is currently unknown. Low-affinity Fst:Bmp interactions are known to be reversible, whereas high-affinity Fst:Activin interactions are effectively irreversible (Iemura et al., 1998). If Fst:Tgfb2 associate irreversibly then spontaneous pattern formation is still possible, but it changes assumptions regarding effective diffusion rates: Fst:Tgfb2 diffusion would then become irrelevant and, instead, Tgfb2 dimers must diffuse faster than Fst monomers (Murray, 2003).

Self-organisation versus positional information *in vivo*
By demonstrating how *Pax6* may drive self-organisation of the primary retinal axis, our findings offer the first mechanistic explanation of the long-known but poorly understood master control function of *Pax6*. In the embryo, we propose that this putative Turing network acts to self-organise the proximal-distal axis of the optic vesicle (as summarised in Fig. 7A,B), in concert with positional information (e.g. from previously identified inductive interactions) to ensure correct alignment with neighbouring tissues.

In Model D, we accounted for intrinsic positional information by incorporating direct suppression of *Pax6* expression by a ventral-high to
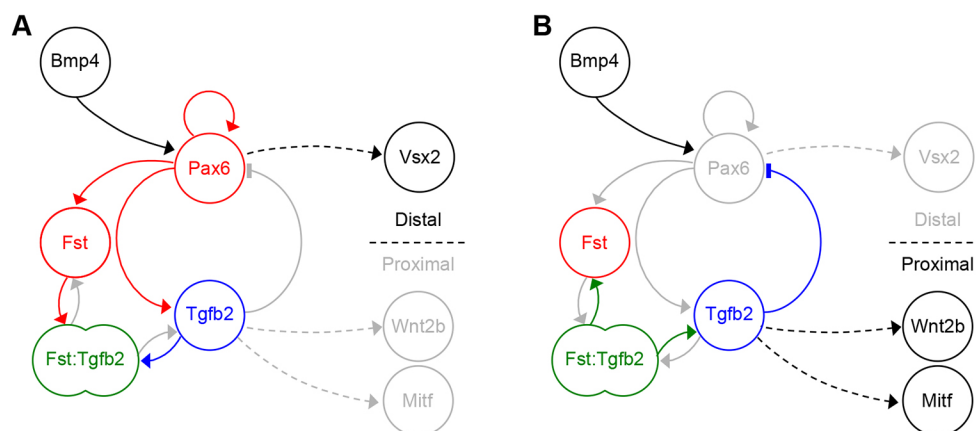
Fig. 7. Proposed *Pax6/Fst/Tgfb2* network function during optic vesicle polarisation *in vivo*. (A) At the prospective distal pole, *Pax6* expression is promoted by upstream Bmp and reinforced via auto-regulation. Pax6 drives distal expression of *Fst*, *Tgfb2* and downstream *Vsx2*. A molar excess of slow-diffusing Fst over Tgfβ receptors is postulated to reversibly sequester Tgfβ2 into fast-diffusing Fst:Tgfβ2 complexes. (B) At the prospective proximal vesicle, dissociation of fast-diffusing Fst:Tgfβ2 complexes is postulated to release Tgfβ2. A molar excess of Tgfβ receptors over slow-diffusing Fst then permits receptor activation by Tgfβ2, causing functional inhibition of Pax6 and induction of proximal markers *Wnt2b* and *Mitf*. Interactions indicated by broken lines may be indirect.

DEVELOPMENT

dorsal-low gradient of Shh activity (Fig. 5; see supplementary information) (Ekker et al., 1995; Macdonald et al., 1995). This is a convenient abstraction, however; at later stages, the ventral extent of *Pax6* expression *in vivo* is refined via reciprocal inhibition between distal *Pax6* (prospective neural retina) and ventral *Pax2* (prospective optic stalk) (Schwarz et al., 2000), the expression of which is activated by ventral Shh (Ekker et al., 1995; Macdonald et al., 1995).

Regarding extrinsic positional information, Bmp signals from the overlying head ectoderm appear to activate the *Pax6/Fst/Tgfb2* network and may also bias proximal-distal polarity to align the distal Pax6$^+$ pole with the prospective lens. This would explain why Bmps from the head ectoderm have been attributed with inducing both proximal retinal pigment epithelium (Müller et al., 2007) and distal neural retina (Pandit et al., 2015) within the optic vesicle.

We did not investigate the role of Wnt in establishing proximal identity within the optic vesicle, except to test for direct synergism between Wnt and Bmp, as previously proposed (Steinfeld et al., 2013). In the absence of such synergism, we suggest that Wnt acts downstream of the *Pax6/Fst/Tgfb2* gene network, as: (1) *Wnt2b* is a Tgfβ target gene (Fuhrmann et al., 2000) restricted to the proximal optic vesicle (Fig. 1G); and (2) expression of *Wnt2b* is absent from the peri-ocular surface ectoderm until stage HH11 (Grocott et al., 2011), prior to which polarised *Wnt2b* expression is already established within the optic vesicle itself (Fig. 1G).

In addition to the loss of inductive signals, ablation of the overlying lens ectoderm (Steinfeld et al., 2013) may permit periocular Tgfβ proteins from the surrounding neural crest mesenchyme (Fuhrmann et al., 2000; Grocott et al., 2011) to overwhelm the autonomous polarising activity of the *Pax6/Fst/Tgfb2* network. In turn, it has not escaped our attention that distal *Fst* may mediate classical lens induction (Spemann, 1901) by opposing these same lens-inhibitory Tgfb signals (Grocott et al., 2011); indeed, *Fst* overexpression induces lens crystallin expression in *Xenopus* animal cap explants (Altmann et al., 1997).

Retinal organoids and self-organisation *in vitro*
During retinal organoid development *in vitro*, we propose that the *Pax6/Fst/Tgfb2* network may suffice to self-organise the primary axis of the retina in the absence of the well-organised positional information normally present *in vivo*. For example, ventral optic vesicle structures are absent in self-organising retinal organoids (Eiraku et al., 2011), which suggests an absence of intrinsic Shh positional information.

The comparatively chaotic nature of organoids makes them an ideal counterpart to embryonic models of development as they can unmask cryptic self-organising mechanisms and test them to breaking point; contrast the straightforward elaboration of an existing pre-pattern (Fig. 4B,B′; analogous to localised *Pax6* induction by neighbouring Bmps *in vivo*) with the more turbulent emergence of order from disorder (Fig. 4D,D′; analogous to spontaneous *Pax6* activation in retinal organoids).

In simulations of *de novo* pattern formation, the *Pax6/Fst/Tgfb2* network is observed to oscillate (Fig. 4D,D′; Movie 2). This potential for oscillation derives from the Eigenvalues associated with the Turing condition and thus from the models' governing equations and parameter choices. For example, in Model B, the tendency to oscillate may be suppressed by increasing the negative feedback that Tgfb2 exerts on Pax6. Whether or not oscillations manifest in a given simulation is further influenced by the choice of initial conditions. For example, Model B is observed to oscillate during *de novo* pattern formation (Fig. 4D,D′; Movie 2), but not when elaborating an existing pre-pattern (equivalent to the Model A

simulation in Fig. 4B,B′; Movie 1). For this reason, we might expect that oscillations are more likely to arise during *de novo* pattern formation in retinal organoid cultures and less so in the embryo, where the wealth of positional information constrains the *Pax6/Fst/Tgfb2* network. Whether or not this gene network oscillates *in vitro* or *in vivo*, and the potential impact on robustness and reproducibility of organoid cultures, is yet to be investigated.

Future directions
A future challenge will be to develop a full 3D model of optic vesicle patterning, incorporating the *Pax6/Fst/Tgfb2* Turing network together with all sources of constraining positional information. A multi-scale approach, in which the feedback between tissue patterning (via the reaction-diffusion formalism used here) and cell dynamics (e.g. via Cellular Potts, vertex or finite element approaches) could further illuminate the feedback between tissue patterning and morphogenesis. A 2D vertex model of optic cup morphogenesis has been previously reported (Eiraku et al., 2012), but a multi-scale approach will be required to fully grasp how genes determine geometry and to identify causal links between genetic and anatomical aberrations.

The identification of defined, animal-free substrates for organoid cultures is a prerequisite for clinical applications. This, and enhanced reproducibility, strongly motivate the search for alternatives to incompletely defined and animal-derived Matrigel, which has superseded laminin as the substrate of choice for *in vitro* retinogenesis (Capowski et al., 2019; Eiraku et al., 2011; Meyer et al., 2009). Interestingly, Matrigel's sixth most abundant ECM component, fibronectin (Rijal and Li, 2017), is enriched within the basement membrane of the optic vesicle *in vivo* (Krotoski et al., 1986; Kurkinen et al., 1979) and binds Fst (Maguer-Satta et al., 2006). Could ECM components such as fibronectin support self-organisation by limiting Fst diffusion relative to Tgfb2 or the Fst: Tgfb2 complex? Further studies are needed to characterise diffusion of these morphogens both *in vivo* and *in vitro*, and to clarify the role of ECM composition in supporting their differential diffusion.

Further exploration of the *Pax6/Fst/Tgfb2* network may drive future developments in retinal organoid technology and help underpin applications in disease modelling, drug discovery and regenerative therapies. Given the deeply homologous nature of the *Pax6* master control function, we would predict that *Pax6* orthologues participate in functionally homologous Turing networks in non-vertebrates, which may comprise the same or different morphogens.

MATERIALS AND METHODS
Chick embryos
Fertile brown hen's eggs (Henry Stewart) were incubated at 38°C in a humidified incubator until the required stage of development: HH8 for *in ovo* electroporation experiments; HH10 for *in vitro* explant experiments. The study was approved by the Animal Welfare & Ethical Review Board, School of Biological Sciences of the University of East Anglia, and all procedures were performed in accordance with the relevant guidelines and regulations.

Explant assays
HH10 embryos were incubated with 0.25% Trypsin-EDTA at 38°C for 7 min. Trypsin was then de-activated by transferring into 20% chick serum on ice for 5 min. Embryos were then washed with Tyrode's solution and pinned onto Sylgard-coated dissection dishes. Head surface ectoderm and peri-ocular mesenchyme were carefully removed using 30 gauge syringe needles from both dorsal and ventral sides. Once cleaned, both optic vesicles were removed and held in Tyrode's solution on ice. Left and right optic vesicles were separately pooled from at least five embryos, yielding two

match-paired pools for use as treated and control samples. Pooled vesicles were cultured in polyHEMA (Sigma)-coated culture wells to prevent adhesion, with DMEM-F12 media (Invitrogen) supplemented with 1× N2 (Invitrogen), 1× L-glutamate and 1× penicillin/streptomycin at 37°C and 5% CO_2 for 16 h. Culture media for treated samples was supplemented with the following factors as required: 35 ng/ml Bmp4 (R&D Systems), 0.5 µM BIO (Sigma) with 0.1% DMSO (Sigma), 10 µM SIS3 with 0.1% DMSO (Sigma), or 2.5 µM cyclopamine (Sigma) with 0.1% 2-hydroxypropyl-β-cyclodextrin (HBC; Sigma).

Whole-mount immunofluorescence staining of explants
Cultured explants were fixed in 4% PFA at 4°C for 90 min, dehydrated and rehydrated through methanol series. After blocking overnight at 4°C in PBTS (BSA, Triton X-100 and goat serum), explants were incubated in mouse anti-Pax6 primary antibody (diluted 1:50 in PBTS; Developmental Studies Hybridoma Bank #PAX6) for 3 days then washed in PBS-Tween. Explants were then incubated in goat anti-mouse Alexa568-conjugated secondary antibody (diluted 1:1000 in PBTS; Life Technologies A-11004) and DAPI for 3 days at 4°C, then washed in PBS-Tween. Stained explants were mounted in AF1 mounting medium (Citifluor) and z-stack images were generated using a Zeiss LSM980 confocal instrument. Relative quantification of nuclear Pax6 fluorescence was performed by normalising to DAPI using the Atlas Toolkit plug-in for FIJI/ImageJ (Grocott et al., 2016) as described.

Quantitative RT-PCR
Explant samples were lysed in 1 ml Trizol (Ambion) and processed for total RNA extraction. RNA samples were digested with DNase I (Ambion) and re-extracted by acidic phenol/chloroform. RNA concentrations were determined by NanoDrop ND-1000 Spectrophotometer. For each experiment, equal quantities of treated and control sample RNA (typically between 0.1–0.6 µg) were used as a template for first strand cDNA synthesis using Superscript II reverse transcriptase (Invitrogen) and random hexamers. cDNAs were diluted 1:20 before relative quantitation of transcript levels by real-time PCR using SYBR Green master mix (Applied Biosystems) and target-specific primers (Table S1). Relative transcript quantification was via the standard curve method, and target gene expression was normalised to the reference gene β-Actin. Fold changes were calculated for each matched-pair (treated/control) then log-transformed to bring data closer to a normal distribution (verified by a Shapiro–Wilk test) prior to plotting and null hypothesis significance testing. These were plotted as mean±s.d. Student's paired t-test was used to calculate the probability of the observed (or more extreme) differences between match-paired (treated and control) sample means, assuming that the null hypothesis is true.

Morpholino knockdown validation
Fst-expressing somite tissue from wild-type chick embryos was dissected and cultured in Dulbecco's modified Eagle medium, 10% foetal bovine serum and 1% penicillin/streptomycin for 4 h before transfecting with 1 mM translation-blocking FstMO (Gene Tools; sequence 5′-GATCCTCTGAT-TTAACATCCTCAGC-3′) or 1 mM StdMO negative control (Gene Tools; sequence 5′-CCTCTTACCTCAGTTACAATTTATA-3′) using Endoporter PEG (Gene Tools). Protein was extracted after 48 h. Protein lysate (30 µg) was run on pre-cast 4-15% polyacrylamide gels (Bio-Rad) and blotted onto polyvinylidene fluoride membrane (Bio-Rad). Primary antibody against Fst (Abcam ab47941; 1:2000) was applied at 4°C overnight and secondary polyclonal goat anti-rabbit-HRP (Cell Signaling Technology, 7074; 1:2000) was applied for 1 h at room temperature. Primary antibody against HSC70 (Santa Cruz, sc-7298; 1:2500) was applied at 4°C overnight and secondary polyclonal goat anti-mouse-HRP (Agilent, P0447; 1:1000) was applied for 1 h at room temperature. The blots were treated with an ECL substrate kit and imaged.

In ovo embryo electroporation
Over-expression constructs encoding dnPax6 and Fst were constructed in pCIG (see Table S3 for PCR primers). Plasmid DNA (2-5 µg/µl) or plasmid DNA and FITC-labelled morpholino oligonucleotides (2 µg/µl and 0.5 mM, respectively) were injected into the open neural tube of stage HH8 chick

embryos in ovo (Fig. 2A). A pair of platinum electrodes connected to an Ovodyne electroporator and current amplifier (Intracel) were then used to electroporate the DNA or DNA+morpholino into either the left or right side of the anterior neural tube via four pulses of 22 V and 50 ms duration at 1 s intervals. Once electroporated, embryos were sealed with adhesive tape and incubated for 10-12 h at 38°C until embryos had reached stage HH10.

Whole-mount in situ hybridisation and immunofluorescence on sections
Embryos were fixed in 4% PFA overnight at 4°C, then dehydrated by methanol series and stored at −20°C. Following re-hydration, embryos were processed for whole-mount in situ hybridisation using 1 µg/ml DIG-labelled antisense probes for Pax6 N-term (Goulding et al., 1993), Pax6 C-term, Vsx2, Mitf, Fst (see Table S2 for PCR primers), Tgfb2 (EST clone ChEST262a17) (Boardman et al., 2002), Wnt2b (a gift from Susan Chapman, Clemson University, SC, USA) and Bmp4 (a gift from Elisa Martí, Institut de Biologia Molecular de Barcelona, Spain). Probes were hybridized at 65°C for up to 72 h. After incubation with 1:5000 anti-DIG antibody (Roche) and washing, 4.5 µl nitroblue tetrazolium (50 mg/ml) and 3.5 µl 5-bromo-4-chloro-3-indolyl phosphate (50 mg/ml) per 1.5 ml developing solution were used for colour development. Embryos were embedded in 7.5% gelatin, 15% sucrose and cryosectioned at 15 µm. Differences in morphology of sections are due to: (1) slight differences in staging of embryos between HH10− and HH10+; and (2) slight obliqueness and variation in the dorsal-ventral level of the horizontal sections. Following de-gelatinisation, sections were blocked in PBTS buffer (PBS with 2% BSA, 0.1% Triton X-100 and 10% goat serum) for 1 h at room temperature. EGFP transgene expression was then detected using rabbit anti-GFP primary antibody (Abcam, ab290; 1:500 dilution) and Alexa568 goat anti-rabbit secondary antibody (Invitrogen, A11036; 1:1000 dilution). Morpholino FITC fluorescence was observed directly. Labelled sections were imaged using a 20× objective on an Axioplan widefield fluorescence microscope with Axiocam HRc camera and Axiovision software (Carl Zeiss).

Relative quantification of in situ hybridisation staining
Assuming that average cell size is invariant between left and right optic vesicles of the same embryo, then the relative area of staining is proportional to the relative number of cells exceeding a common detection threshold. To quantify this, bright-field micrographs were converted to greyscale, inverted then thresholded and the area of optic vesicle staining measured in FIJI (Schindelin et al., 2012). Transfected and contralateral controls from the same embryo were processed simultaneously to ensure identical treatment. Staining area in transfected vesicles was then normalised to internal contralateral controls, yielding a fold-change in gene expression area. Fold changes were log-transformed to bring data closer to a normal distribution (verified by Shapiro–Wilk test) prior to plotting and null hypothesis significance testing. Box plots showing mean Log10(fold change)±s.d. were generated in R with the package 'Beeswarm'. A Welch's two-sample t-test (for pairwise comparisons) or one-way ANOVA with Tukey's post-hoc test (for groupwise comparisons) were used to calculate the probability of the observed (or more extreme) differences between sample means assuming that the null hypothesis is true.

Reaction-diffusion simulations
Partial differential equations were coded in R using the function tran.1d() and tran.2d() from package 'ReacTran' to handle diffusion terms. 1D and 2D numerical simulations used the functions ode.1d() and ode.2d(), respectively, from package 'deSolve' and the default integrator. Parameter sweeps were performed to identify suitable diffusion rates (see Movies 1 and 2). 1D simulations were run with both periodic and zero-flux boundary conditions, with comparable results. 2D simulations were performed with zero-flux boundary conditions on explant-shaped domains, which best reflected experimental observations. See supplementary information for model code and narrative text. The model code is explained in the supplementary information, is available via our GitHub repository (https://github.com/GrocottLab/) and is accessible as an interactive Jupyter Notebook (https://mybinder.org/v2/gh/GrocottLab/Pax6-Fst-Tgfb2_Reaction_Diffusion_Models/master).

Data availability
The model code is explained in the supplementary information, is available via our GitHub repository (https://github.com/GrocottLab/) and is accessible as an interactive Jupyter Notebook (https://mybinder.org/v2/gh/GrocottLab/Pax6-Fst-Tgfb2_Reaction_Diffusion_Models/master).

Supplementary information
Supplementary information available online at
https://dev.biologists.org/lookup/doi/10.1242/dev.185827.supplemental

Peer review history
The peer review history is available online at
https://dev.biologists.org/lookup/doi/10.1242/dev.185827.reviewer-comments.pdf

References
Altmann, C. R., Chow, R. L., Lang, R. A. and Hemmati-Brivanlou, A. (1997). Lens induction by Pax-6 in Xenopus laevis. *Dev. Biol.* **185**, 119-123. doi:10.1006/dbio.1997.8573

Ashery-Padan, R., Marquardt, T., Zhou, X. and Gruss, P. (2000). Pax6 activity in the lens primordium is required for lens formation and for correct placement of a single retina in the eye. *Genes Dev.* **14**, 2701-2711. doi:10.1101/gad.184000

Boardman, P. E., Sanz-Ezquerro, J., Overton, I. M., Burt, D. W., Bosch, E., Fong, W. T., Tickle, C., Brown, W. R. A., Wilson, S. A. and Hubbard, S. J. (2002). A comprehensive collection of chicken cDNAs. *Curr. Biol.* **12**, 1965-1969. doi:10.1016/S0960-9822(02)01296-4

Capowski, E. E., Samimi, K., Mayerl, S. J., Phillips, M. J., Pinilla, I., Howden, S. E., Saha, J., Jansen, A. D., Edwards, K. L., Jager, L. D. et al. (2019). Reproducibility and staging of 3D human retinal organoids across multiple pluripotent stem cell lines. *Development* **146**, dev171686. doi:10.1242/dev.171686

Chen, J. K., Taipale, J., Cooper, M. K. and Beachy, P. A. (2002). Inhibition of Hedgehog signaling by direct binding of cyclopamine to Smoothened. *Genes Dev.* **16**, 2743-2748. doi:10.1101/gad.1025302

Chow, R. L., Altmann, C. R., Lang, R. A. and Hemmati-Brivanlou, A. (1999). Pax6 induces ectopic eyes in a vertebrate. *Development* **126**, 4213-4222.

Cvekl, A. and Callaerts, P. (2017). PAX6: 25th anniversary and more to learn. *Exp. Eye Res.* **156**, 10-21. doi:10.1016/j.exer.2016.04.017

Eiraku, M., Takata, N., Ishibashi, H., Kawada, M., Sakakura, E., Okuda, S., Sekiguchi, K., Adachi, T. and Sasai, Y. (2011). Self-organizing optic-cup morphogenesis in three-dimensional culture. *Nature* **472**, 51-56. doi:10.1038/nature09941

Eiraku, M., Adachi, T. and Sasai, Y. (2012). Relaxation-expansion model for self-driven retinal morphogenesis: a hypothesis from the perspective of biosystems dynamics at the multi-cellular level. *BioEssays* **34**, 17-25. doi:10.1002/bies.201100070

Ekker, S. C., Ungar, A. R., Greenstein, P., von Kessler, D. P., Porter, J. A., Moon, R. T. and Beachy, P. A. (1995). Patterning activities of vertebrate hedgehog proteins in the developing eye and brain. *Curr. Biol.* **5**, 944-955. doi:10.1016/S0960-9822(95)00185-0

Esteve, P., Sandonis, A., Ibañez, C., Shimono, A., Guerrero, I. and Bovolenta, P. (2011). Secreted frizzled-related proteins are required for Wnt/β-catenin signalling activation in the vertebrate optic cup. *Development* **138**, 4179-4184. doi:10.1242/dev.065839

Fuhrmann, S., Levine, E. M. and Reh, T. A. (2000). Extraocular mesenchyme patterns the optic vesicle during early eye development in the embryonic chick. *Development* **127**, 4599-4609.

Gehring, W. J. (1996). The master control gene for morphogenesis and evolution of the eye. *Genes Cells* **1**, 11-15. doi:10.1046/j.1365-2443.1996.11011.x

Gierer, A. and Meinhardt, H. (1972). A theory of biological pattern formation. *Kybernetik* **12**, 30-39. doi:10.1007/BF00289234

Goulding, M. D., Lumsden, A. and Gruss, P. (1993). Signals from the notochord and floor plate regulate the region-specific expression of two Pax genes in the developing spinal cord. *Development* **117**, 1001-1016.

Grocott, T., Frost, V., Maillard, M., Johansen, T., Wheeler, G. N., Dawes, L. J., Wormstone, I. M. and Chantry, A. (2007). The MH1 domain of Smad3 interacts with Pax6 and represses autoregulation of the Pax6 P1 promoter. *Nucleic Acids Res.* **35**, 890-901. doi:10.1093/nar/gkl1105

Grocott, T., Johnson, S., Bailey, A. P. and Streit, A. (2011). Neural crest cells organize the eye via TGF-β and canonical Wnt signalling. *Nat. Commun.* **2**, 265. doi:10.1038/ncomms1269

Grocott, T., Thomas, P. and Münsterberg, A. E. (2016). Atlas Toolkit: Fast registration of 3D morphological datasets in the absence of landmarks. *Sci. Rep.* **6**, 20732. doi:10.1038/srep20732

Gunhaga, L. (2011). The lens: a classical model of embryonic induction providing new insights into cell determination in early development. *Phil. Trans. R. Soc. B* **366**, 1193-1203. doi:10.1098/rstb.2010.0175

Halder, G., Callaerts, P. and Gehring, W. J. (1995). Induction of ectopic eyes by targeted expression of the eyeless gene in Drosophila. *Science* **267**, 1788-1792. doi:10.1126/science.7892602

Hamburger, V. and Hamilton, H. L. (1992). A series of normal stages in the development of the chick embryo. 1951. *Dev Dyn* **195**, 231-272. doi:10.1002/aja.1001950404

Hasegawa, Y., Takata, N., Okuda, S., Kawada, M., Eiraku, M. and Sasai, Y. (2016). Emergence of dorsal-ventral polarity in ESC-derived retinal tissue. *Development* **143**, 3895-3906. doi:10.1242/dev.134601

Hemmati-Brivanlou, A., Kelly, O. G. and Melton, D. A. (1994). Follistatin, an antagonist of activin, is expressed in the Spemann organizer and displays direct neuralizing activity. *Cell* **77**, 283-295. doi:10.1016/0092-8674(94)90320-4

Hill, R. E., Favor, J., Hogan, B. L. M., Ton, C. C. T., Saunders, G. F., Hanson, I. M., Prosser, J., Jordan, T., Hastie, N. D. and van Heyningen, V. (1991). Mouse Small eye results from mutations in a paired-like homeobox-containing gene. *Nature* **354**, 522-525. doi:10.1038/354522a0

Hodgson, S. V. and Saunders, K. E. (1980). A probable case of the homozygous condition of the aniridia gene. *J. Med. Genet.* **17**, 478-480. doi:10.1136/jmg.17.6.478

Hoge, M. A. (1915). Another gene in the fourth chromosome of Drosophila. *Am. Nat.* **49**, 47-49. doi:10.1086/279455

Iemura, S.-I., Yamamoto, T. S., Takagi, C., Uchiyama, H., Natsume, T., Shimasaki, S., Sugino, H. and Ueno, N. (1998). Direct binding of follistatin to a complex of bone-morphogenetic protein and its receptor inhibits ventral and epidermal cell fates in early Xenopus embryo. *Proc. Natl. Acad. Sci. USA* **95**, 9337-9342. doi:10.1073/pnas.95.16.9337

Jinnin, M., Ihn, H. and Tamaki, K. (2006). Characterization of SIS3, a novel specific inhibitor of Smad3, and its effect on transforming growth factor-β1-induced extracellular matrix expression. *Mol. Pharmacol.* **69**, 597-607. doi:10.1124/mol.105.017483

Krotoski, D. M., Domingo, C. and Bronner-Fraser, M. (1986). Distribution of a putative cell surface receptor for fibronectin and laminin in the avian embryo. *J. Cell Biol.* **103**, 1061-1071. doi:10.1083/jcb.103.3.1061

Kurkinen, M., Alitalo, K., Vaheri, A., Stenman, S. and Saxén, L. (1979). Fibronectin in the development of embryonic chick eye. *Dev. Biol.* **69**, 589-600. doi:10.1016/0012-1606(79)90313-0

Liu, I. S. C., Chen, J.-D., Ploder, L., Vidgen, D., van der Kooy, D., Kalnins, V. I. and McInnes, R. R. (1994). Developmental expression of a novel murine homeobox gene (Chx10): evidence for roles in determination of the neuroretina and inner nuclear layer. *Neuron* **13**, 377-393. doi:10.1016/0896-6273(94)90354-9

Macdonald, R., Barth, K. A., Xu, Q., Holder, N., Mikkola, I. and Wilson, S. W. (1995). Midline signalling is required for Pax gene regulation and patterning of the eyes. *Development* **121**, 3267-3278.

Maguer-Satta, V., Forissier, S., Bartholin, L., Martel, S., Jeanpierre, S., Bachelard, E. and Rimokh, R. (2006). A novel role for fibronectin type I domain in the regulation of human hematopoietic cell adhesiveness through binding to follistatin domains of FLRG and follistatin. *Exp. Cell Res.* **312**, 434-442. doi:10.1016/j.yexcr.2005.11.006

Massagué, J. (1998). TGF-β signal transduction. *Annu. Rev. Biochem.* **67**, 753-791. doi:10.1146/annurev.biochem.67.1.753

Meijer, L., Skaltsounis, A.-L., Magiatis, P., Polychronopoulos, P., Knockaert, M., Leost, M., Ryan, X. P., Vonica, C. A., Brivanlou, A., Dajani, R. et al. (2003). GSK-3-selective inhibitors derived from Tyrian purple indirubins. *Chem. Biol.* **10**, 1255-1266. doi:10.1016/j.chembiol.2003.11.010

Meyer, J. S., Shearer, R. L., Capowski, E. E., Wright, L. S., Wallace, K. A., McMillan, E. L., Zhang, S.-C. and Gamm, D. M. (2009). Modeling early retinal development with human embryonic and induced pluripotent stem cells. *Proc. Natl. Acad. Sci. USA* **106**, 16698-16703. doi:10.1073/pnas.0905245106

DEVELOPMENT

Müller, F., Rohrer, H. and Vogel-Höpker, A. (2007). Bone morphogenetic proteins specify the retinal pigment epithelium in the chick embryo. *Development* **134**, 3483-3493. doi:10.1242/dev.02884

Murray, J. D. (2003). *Mathematical Biology. II Spatial Models and Biomedical Applications*. New York: Springer-Verlag. https://doi.org/10.1007/b98869

Nakamura, T., Takio, K., Eto, Y., Shibai, H., Titani, K. and Sugino, H. (1990). Activin-binding protein from rat ovary is follistatin. *Science* **247**, 836-838. doi:10.1126/science.2106159

Nakamura, T., Sugino, K., Titani, K. and Sugino, H. (1991). Follistatin, an activin-binding protein, associates with heparan sulfate chains of proteoglycans on follicular granulosa cells. *J. Biol. Chem.* **266**, 19432-19437.

Nakayama, T., Fisher, M., Nakajima, K., Odeleye, A. O., Zimmerman, K. B., Fish, M. B., Yaoita, Y., Chojnowski, J. L., Lauderdale, J. D., Netland, P. A. et al. (2015). Xenopus pax6 mutants affect eye development and other organ systems, and have phenotypic similarities to human aniridia patients. *Dev. Biol.* **408**, 328-344. doi:10.1016/j.ydbio.2015.02.012

Nogai, H., Rosowski, M., Grün, J., Rietz, A., Debus, N., Schmidt, G., Lauster, C., Janitz, M., Vortkamp, A. and Lauster, R. (2008). Follistatin antagonizes transforming growth factor-β3-induced epithelial–mesenchymal transition in vitro: implications for murine palatal development supported by microarray analysis. *Differentiation* **76**, 404-416. doi:10.1111/j.1432-0436.2007.00223.x

Pandit, T., Jidigam, V. K., Patthey, C. and Gunhaga, L. (2015). Neural retina identity is specified by lens-derived BMP signals. *Development* **142**, 1850-1859. doi:10.1242/dev.123653

Rijal, G. and Li, W. (2017). A versatile 3D tissue matrix scaffold system for tumor modeling and drug screening. *Sci. Adv.* **3**, e1700764. doi:10.1126/sciadv.1700764

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B. et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Meth.* **9**, 676-682. doi:10.1038/nmeth.2019

Schwarz, M., Cecconi, F., Bernier, G., Andrejewski, N., Kammandel, B., Wagner, M. and Gruss, P. (2000). Spatial specification of mammalian eye territories by reciprocal transcriptional repression of Pax2 and Pax6. *Development* **127**, 4325-4334.

Shubin, N., Tabin, C. and Carroll, S. (1997). Fossils, genes and the evolution of animal limbs. *Nature* **388**, 639-648. doi:10.1038/41710

Spemann, H. (1901). Uber Korrelationen in der Entwicklung des Auges. *Verh. Anat. Ges.* **15**, 61-79.

Steinfeld, J., Steinfeld, I., Coronato, N., Hampel, M.-L., Layer, P. G., Araki, M. and Vogel-Höpker, A. (2013). RPE specification in the chick is mediated by surface ectoderm-derived BMP and Wnt signalling. *Development* **140**, 4959-4969. doi:10.1242/dev.096990

Tomarev, S. I., Callaerts, P., Kos, L., Zinovieva, R., Halder, G., Gehring, W. and Piatigorsky, J. (1997). Squid Pax-6 and eye development. *Proc. Natl. Acad. Sci. USA* **94**, 2421-2426. doi:10.1073/pnas.94.6.2421

Turing, A. M. (1952). The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **237**, 37-72. doi:10.1098/rstb.1952.0012

Wolf, L. V., Yang, Y., Wang, J., Xie, Q., Braunger, B., Tamm, E. R., Zavadil, J. and Cvekl, A. (2009). Identification of pax6-dependent gene regulatory networks in the mouse lens. *PLoS ONE* **4**, e4159. doi:10.1371/journal.pone.0004159

DEVELOPMENT

Supplementary Information

1D simulation of Reaction-Diffusion Model A:

Introduction:

In this model, Pax6 drives expression of both Fst and Tgfb2. Fst monomers and Tgfb2 dimers may associate to form a hetero-tetrameric Fst:Tgfb2 complex. The Fst:Tgfb2 complex may dissociate in turn, thereby releasing Fst monomers and Tgfb2 dimers. Tgfb2 signals via association with Type I and Type II receptors, yielding an activated hetero-hexameric Tgfb2:Tgfbr signalling complex (comrpising one Tgfb2 ligand dimer and four Tgfb receptors - two Type I and two Type II; not shown here).
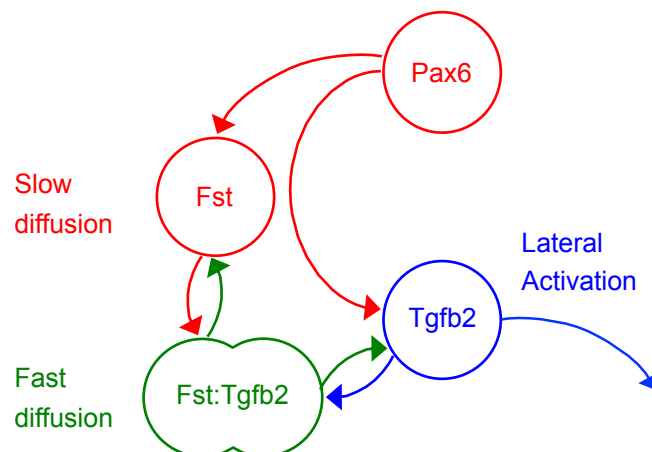
Figure S1. Summary of Model A in which Pax6 drives expression of both Fst and Tgfb2, whereas Fst inhibits Tgfb2 function via sequestration. Slow diffusion of Fst was postulated to result in local inhibition of Tgfb2 at the source of Pax6/Tgfb2/Fst expression. Conversely, fast diffusion of Tgfb2 was postulated to drive lateral activation of its downstream signalling pathway away from the Pax6/Fst/Tgfb2-expressing region.

As dicussed in the main article, Fst monomers are assumed to exhibit slower effective diffusion than Fst:Tgfb2 complexes, whereas effective diffusion of Tgfb2 dimers relative to Fst:Tgfb2 complexes is less critical. If this differential diffusion condition is met, Model A is able to elaborate an existing Pax6 pre-pattern (e.g. expression induced within the distal optic vesicle by upstream BMP signalling) via 'lateral activation' of Tgfb receptor complexes away from the source of Pax6 expression (e.g. within the adjacent proximal optic vesicle). In other words, the pattern of Tgfb pathway activation becomes the inverse of upstream Pax6 expression.

Equations:

The following system of equations formally represents the above interactions. Each equation describes the rate of change of concentration for the various molecular species downstream of Pax6. The initial concentrations of Pax6 (P) and Tgfb receptors (R) are held constant throughout the simulation, i.e. their rates of change, $\frac{\partial[P]}{\partial t}$ and $\frac{\partial[R]}{\partial t}$ are set to zero throughout.

Fst:

$$\frac{\partial[F]}{\partial t} = a_F . \frac{[P]}{[P]+1} - 2.a_{FT} . [F]^2 . [T] + 2.b_{FT} . [FT] - b_F . [F] + D_F \frac{d^2[F]}{dx^2}$$

Where,

- Pax6 (P) positively regulates Fst expression via Michaelis-Menton kinetics $\frac{[P]}{[P]+1}$.
- Pairs of Fst monomers are consumed in the formation of Fst:Tgfb2 complexes (FT) via $-2.a_{FT} . [F]^2 . [T]$, and released again via $+2.b_{FT} . [FT]$. This implements the two-way reaction: $1\,\text{T} + 2\,\text{F} \underset{b_{FT}}{\overset{a_{FT}}{\rightleftharpoons}} 1\,\text{FT}$, whereby one Tgfb2 dimer reversibly associates with two Fst monomers.
- Fst is turned over via $-b_F . [F]$.
- Fst diffuses via $D_F \frac{d^2[F]}{dx^2}$.

Tgfb2:

$$\frac{\partial[T]}{\partial t} = \left(a_T . \frac{[P]}{[P]+1} \right)^2 - a_{FT} . [F]^2 . [T] + b_{FT} . [FT] - a_{RT} . [R]^4 . [T] - b_T . [T] + D_T \frac{d^2[T]}{dx^2}$$

Where,

- Pax6 (P) positively regulates Tgfb2 expression via Michaelis-Menton kinetics $\frac{[P]}{[P]+1}$.

- Tgfb2 is produced/secreted as a disulphide linked dimer, hence $\left(a_T . \frac{[P]}{[P]+1} \right)^2$.
- Tgfb2 is sequestered in the formation of Fst:Tgfb2 complexes (FT) via $-a_{FT} . [F]^2 . [T]$, and released again via $+b_{FT} . [FT]$. This implements the two-way reaction: $1\,\text{T} + 2\,\text{F} \underset{b_{FT}}{\overset{a_{FT}}{\rightleftharpoons}} 1\,\text{FT}$, whereby one Tgfb2 dimer reversibly associates with two Fst monomers.
- Tgfb2 is irreversibly consumed in the formation of activated receptor complexes (RT) via $-a_{RT} . [R]^4 . [T]$. This implements the one-way reaction: $1\,\text{T} + 4\,\text{R} \overset{a_{RT}}{\longrightarrow} 1\,\text{RT}$, whereby one Tgfb2 dimer associates with four Tgfbr monomers (two type I receptors + two type II receptors). This reaction is irreverible since cells internalise and ultimately turn over activated receptor complexes.
- Tgfb2 is turned over via $-b_T . [T]$.
- Tgfb2 diffuses via $D_T \frac{d^2[T]}{dx^2}$.

Fst:Tgfb2 complex:

$$\frac{\partial[FT]}{\partial t} = a_{FT} . [F]^2 . [T] - b_{FT} . [FT] + D_{FT} \frac{d^2[FT]}{dx^2}$$

Where,

- A pair of Fst monomers and a single Tgfb2 dimer associate via $a_{FT} . [F]^2 . [T]$ forming a Fst:Tgfb2 complex (FT), which dissociates via $-b_{FT} . [FT]$. These terms implement the two-way reaction: $1\,\text{T} + 2\,\text{F} \underset{b_{FT}}{\overset{a_{FT}}{\rightleftharpoons}} 1\,\text{FT}$.
- Fst:Tgfb2 diffuses via $D_{FT} \frac{d^2[FT]}{dx^2}$.

Tgfb2:Tgfbr signalling complex:

$$\frac{\partial[RT]}{\partial t} = a_{RT}.[R]^4.[T] - b_{RT}.[RT]$$

Where,

- Active Tgfb2:Tgfbr signalling complexes (RT) are formed via $a_{RT}.[R]^4.[T]$. This implements the one-way reaction: $1\,T + 4\,R \xrightarrow{a_{RT}} 1\,RT$, whereby one Tgfb2 dimer associates with four Tgfbr monomers (two type I receptors + two type II receptors). This reaction is irreverible since cells internalise and ultimately turn over activated receptor complexes.
- Active Tgfb2:Tgfbr signalling complexes are turned over via $-b_{RT}.[RT]$.

R implementation:

The function 'modelA' implements the above system of equations and calculates the rates of change in concentration for each molecular species. The code breaks down into the following steps:

- Fetch the current system state (i.e. molecular concentrations for each 'cell' in the tissue).

- Evaluate the reaction terms for each molecular species.

- Evaluate diffusion terms for diffusible species. Diffusion terms are written for zero-flux boundary conditions. Terms for periodic boundaries are provided but commented out due to considerably longer execution times.

- Combine reaction and diffusion terms to determine rates of change for each species.

- Update a progress bar.

- Return the list of rates.

```
In [1]:  library(deSolve)
         library(ReacTran)
         library(viridis)
         library(astro)
         library(ggplot2)
         library(readbitmap)

         ## =================
         ## Model equations
         ## =================
         modelA <- function(time, state, parms) {
           with (as.list(parms), {

             # Generalise this: state[ ( (i-1)*N.A + 1) : (i * N.A) ]
             PAX6  <- state[1:N.A]
             TGFB2 <- state[(N.A+1):(2*N.A)]
             FST   <- state[(2*N.A+1):(3*N.A)]
             FT    <- state[(3*N.A+1):(4*N.A)]
             TGFBR <- state[(4*N.A+1):(5*N.A)]
             RT    <- state[(5*N.A+1):(6*N.A)]

             ## Reaction terms:
             reacPAX6  <- rep(0, N.A) # Concentration does not vary throughout simulati
         on
             reacTGFB2 <- (
                           + (aTGFB2 * PAX6 / (PAX6 + 1) ) * (aTGFB2 * PAX6 / (PAX6 + 1
         ) )
                           - (2 * aFT * FST * FST * TGFB2 )
                           + (2 * bFT * FT)
                           - (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                           - (bTGFB2 * TGFB2)
                          )
             reacFST   <- (
                           + (aFST * PAX6) / ( (PAX6 + 1) )
                           - (2 * aFT * FST * FST * TGFB2)
                           + (2 * bFT * FT)
                           - (bFST * FST)
                          )
             reacFT    <- (
                           + (aFT * FST * FST * TGFB2)
                           - (bFT * FT)
                          )
             reacTGFBR <- rep(0, N.A) # Concentration does not vary throughout simulati
         on
             reacRT    <- (
                           + (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                           - (bRT * RT)
                          )

             ## Diffusion terms - periodic boundary
             ## NOTE: these take longer to compute
             #diffTGFB2 <- tran.1D(C = TGFB2,
             #                      C.up = TGFB2[N.A],
             #                      C.down = TGFB2[1],
             #                      D = dTGFB2,
             #                      dx = xgridA)$dC
             #diffFST   <- tran.1D(C = FST,
             #                      C.up = FST[N.A],
             #                      C.down = FST[1],
             #                      D = dFST,
             #                      dx = xgridA)$dC
             #diffFT    <- tran.1D(C = FT,
             #                      C.up = FT[N.A],
             #                      C.down = FT[1],
             #                      D = dFT,
```

```r
    #                         dx = xgridA)$dC

    ## Diffusion terms - zero-flux boundary
    diffTGFB2 <- tran.1D(C = TGFB2,
                         C.up = TGFB2[1],
                         C.down = TGFB2[N.A],
                         D = dTGFB2,
                         dx = xgridA)$dC
    diffFST   <- tran.1D(C = FST,
                         C.up = FST[1],
                         C.down = FST[N.A],
                         D = dFST,
                         dx = xgridA)$dC
    diffFT    <- tran.1D(C = FT,
                         C.up = FT[1],
                         C.down = FT[N.A],
                         D = dFT,
                         dx = xgridA)$dC

    ## Diffusion terms - fixed boundary
    #diffTGFB2 <- tran.1D(C = TGFB2,
    #                     C.up = 0,
    #                     C.down = 0,
    #                     D = dTGFB2,
    #                     dx = xgridA)$dC
    #diffFST   <- tran.1D(C = FST,
    #                     C.up = 0,
    #                     C.down = 0,
    #                     D = dFST,
    #                     dx = xgridA)$dC
    #diffFT    <- tran.1D(C = FT,
    #                     C.up = 0,
    #                     C.down = 0,
    #                     D = dFT,
    #                     dx = xgridA)$dC

    ## Rates of change = reaction  + diffusion:
    deltaPAX6       = reacPAX6
    deltaTGFB2      = reacTGFB2 + diffTGFB2
    deltaFST        = reacFST   + diffFST
    deltaFT         = reacFT    + diffFT
    deltaTGFBR      = reacTGFBR
    deltaRT         = reacRT

    setTxtProgressBar(pb, time)

    return (list(c(deltaPAX6, deltaTGFB2, deltaFST, deltaFT, deltaTGFBR, delta
RT)))
  })
}
```

```
Loading required package: rootSolve

Loading required package: shape

Loading required package: viridisLite

Loading required package: MASS

Loading required package: plotrix

Warning message:
"package 'plotrix' was built under R version 3.5.2"
```

Parameters:

```
In [2]:  ## ===================
         ## Model parameters:
         ## ===================

         ## Space:
         L.A      <- 50                                    # Length of the 1D tissue
         N.A      <- 100                                   # Number of 1D 'cells'
         dxA      <- L.A/N.A                               # Length of one 1D 'cell'
         xgridA <- setup.grid.1D(N = N.A, x.down = L.A) # Output wanted at these cell p
         ositions

         ## Time:
         TimeA    <- 1500                       # Duration of the simulation
         StepsA   <- 40                         # Number of time-steps
         dtA      <-  TimeA/StepsA              # Size of each time-step
         timesA   <- seq(0, TimeA, by = dtA)   # Output wanted at these time intervals

         ## Molecuar Species
         s.names = c("PAX6",
                     "TGFB2 Dimer",
                     "FST Monomer",
                     "FST:TGFB2\nHeterotetramer",
                     "TGFBR Monomer",
                     "TGFB2:TGFBR\nComplex"
                    )
         SPECIES = length(s.names)

         # Setup text progress bar - called by model function
         pbA <- txtProgressBar(min = 0, max = TimeA, style = 3)

         ## Rate constants:
         parmsA <- c(pb     <- pbA,     # Progress bar
                     xgrid  <- xgridA,

                     aPAX6  = 0.3,     # Production rate constant for PAX6
                     bPAX6  = 0.1,     # Decay rate constant for PAX6

                     aTGFB2 = 0.25,    # Production rate constant for TGFB2 dimer
                     bTGFB2 = 0.1,     # Deacy rate constant for TGFB2 dimer
                     dTGFB2 = 1,       # Diffusion rate constant for TGFB2 dimer

                     aFST   = 1.5,     # Production rate constant for FST
                     bFST   = 0.1,     # Decay rate constant for FST
                     dFST   = 1,       # Diffusion rate constant for FST

                     aFT    = 5,       # Association rate constant for FST:TGFB2 comple
         x
                     bFT    = 1,       # Dissociation rate constant for FST:TGFB2 compl
         ex
                     dFT    = 50,      # Diffusion rate constant for FST:TGBF2 complex

                     aRT    = 20,      # Association rate constant for TGFBR:TGFB2 comp
         lex
                     bRT    = 0.1      # Decay rate constant for TGFBR:TGFB2 complex
                    )
```

```
         |                                                                          |
0%
```

Initial Conditions:

The initial conditions describe a situation in which Pax6 expression is restricted to one region within the simulated tissue, corresponding to the distal optic vesicle. This is analagous to the situation in the embryo, where BMPs from the presumptive lens ectoderm promote Pax6 expression in the distal optic vesicle (see main text). Initial distributions of all other molecular species are uniform.

```
In [3]:  ## ===================
         ## Initial conditions:
         ## ===================
         stateA <- c(rep(0, N.A*(1.5/5)), rep(3, N.A*(2/5)), rep(0, N.A*(1.5/5)),   # PA
         X6 pre-pattern
                   rep(0, N.A),                                                      # TG
         FB2
                   rep(0, N.A),                                                      # FS
         T
                   rep(0, N.A),                                                      # FT
                   rep(1, N.A),                                                      # TG
         FBR
                   rep(0, N.A) )                                                     # RT
```

Run Simulation:

```
In [4]:  ## ===================
         # Run the simulation:
         ## ===================
         outA <- ode.1D(y        = stateA,
                        times    = timesA,
                        func     = modelA,
                        parms    = parmsA,
                        maxsteps = 1000000,
                        nspec    = SPECIES,
                        names    = s.names
                        )

         # Close the progress bar
         close(pb)
```

```
  |=====================================================================|  10
0%
```

In [5]:
```r
## =============================
## Plot the output via ggplot2
## =============================
library(ggplot2)
print("Formatting simulation data...")
# Create data.frame for display data
x <- seq(from=dxA,to=N.A*dxA, by=dxA) # Length of tissue
t <- timesA                           # Time steps
s <- s.names                          # Species names
dataA <- expand.grid(X=x, Species=s, Time=t, Concentration= 0.0)

# Read simulation output into data.frame ready for display
# For each species...
for (s in seq(from=1,to=SPECIES,by=1)) {
    # And working backwards for each timepoint...
    for (t in timesA) {
        # Find column range in output for current species (timepoints are in s
eperate rows)
        s.start <- 2 + N.A * ( s - 1 )
        s.end   <- s.start + N.A - 1
        # Extract data for current timepoint (row) and species (columns)
        out.ACT <- outA[1+t/dtA, s.start:s.end]
        # Normalise for this species at this timepoint only
        #out.ACT <- out.ACT / max(out.ACT)
        # Find rows in data.frame for current species and timepoint
        s.index <- dataA$Species == s.names[s] # Rows for species
        t.index <- dataA$Time    == t          # Rows for timeopoint
        # Insert data for current species and timepoint
        dataA[s.index & t.index, 4] <- out.ACT

    }
    # Normalise across all timepoints for each species
    maximum <- max(dataA[s.index, 4])
    dataA[s.index, 4] <- dataA[s.index, 4] / maximum
}
print("...done!")

# Generate multi-faceted plot for endpoint...
print("Rendering endpoint plot...")
just.PAX6 <- dataA$Species == "PAX6"
just.RT   <- dataA$Species == s.names[6]
ggplot(dataA, aes(x=Time, y=X, z=Concentration)) +
geom_tile(aes(fill = Concentration ) ) +
facet_wrap(~ Species, ncol=3) +
theme_bw() +
labs(fill = "Normalised\nConcentration") +
ylab("Anterior-Posterior Axis (AU)") +
xlab("Time (AU)") +
theme(text = element_text(size=12)
      ) +
scale_fill_viridis(option="B")
print("...done!")
```

```
[1] "Formatting simulation data..."
[1] "...done!"
[1] "Rendering endpoint plot..."
[1] "...done!"
```
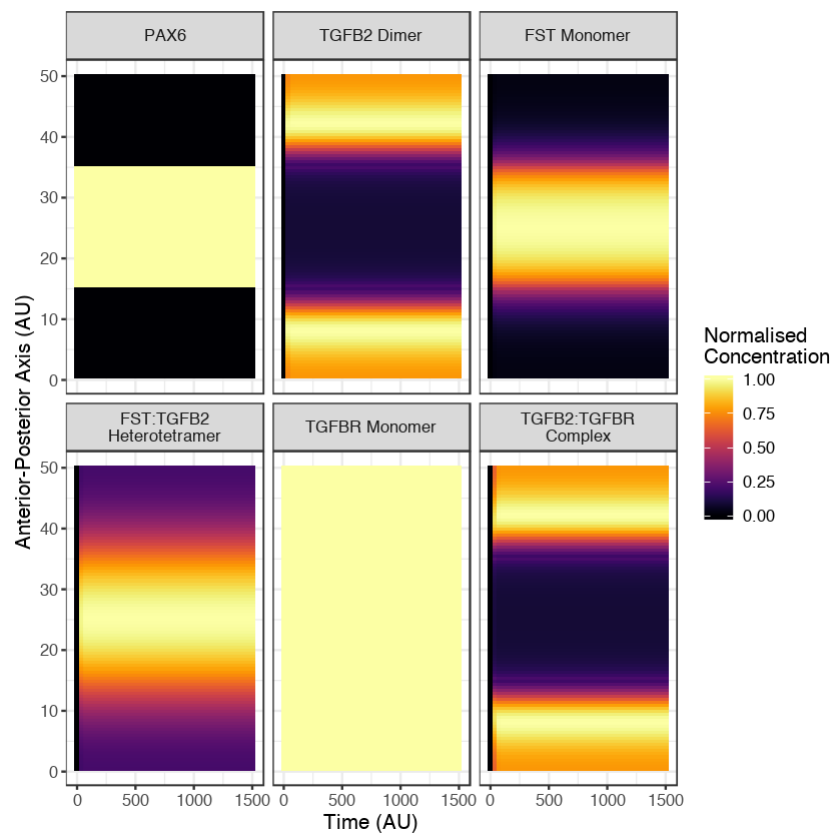

Figure S2. 1-D numerical simulation of Model A showing the normalised concentration distributions of each molecular species evolving over time. Brighter colours = higher concentrations.

1D simulation of Reaction-Diffusion Model B:

Introduction:

Model B is a direct extension of Model A, whereby Tgfb signalling inhibits Pax6's transcriptional activator function.
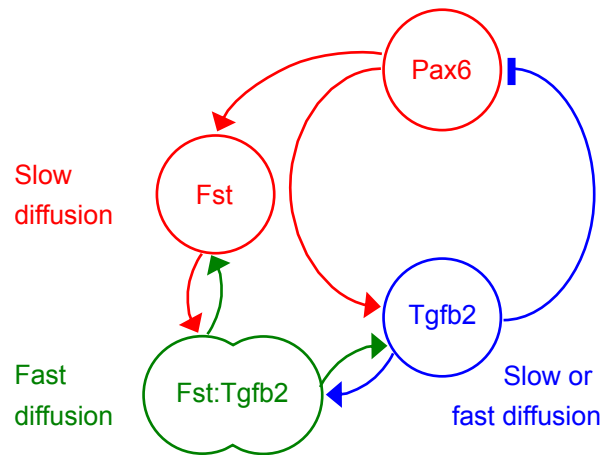
Figure S3. Summary of Model B in which Fst:Tgfb2 complex quickly diffuses and dissociates while Tgfb2 additionally inhibits Pax6 transcriptional activator function.

Note: In the following implementation of Model B Pax6 positively autoregulates its own expression. However, this is not strictly necessary for self-organisation to occur. This is because an indirect positive autoregulatory loop already exists; Pax6 acts via Fst to suppress Tgfb2's inhibition of Pax6 within the distal (*Pax6/Fst/Tgfb2*-expressing) pole. This indirect positive autoregulatory loop ("local autocatalysis") combined with lateral activation of Tgfb receptors ("long-range inhibition" of Pax6 function) is sufficient for self-organisation, though some parameter choices may require adjustment from those presented here.

<u>Equations:</u>

Pax6:

$$\frac{\partial [P]}{\partial t} = a_P . \frac{[P]}{[RT].[P]+1} - b_P . [P]$$

Where,

- Pax6 (P) positively autoregulates via Michaelis-Menton kinetics $\frac{[P]}{[P]+1}$.
- Activated Tgfbr complexes (RT) suppress Pax6 function via $\frac{[P]}{[RT].[P]+1}$.
- Pax6 is turned over via $-b_P . [P]$.

Tgfb2:

$$\frac{\partial [T]}{\partial t} = \left(a_T . \frac{[P]}{[RT].[P]+1} \right)^2 - a_{FT} . [F]^2 . [T] + b_{FT} . [FT] - a_{RT} . [R]^4 . [T] - b_T . [T] + D_T \frac{d^2[T]}{dx^2}$$

Where,

- Pax6 (P) positively regulates Tgfb2 expression via Michaelis-Menton kinetics $\frac{[P]}{[P]+1}$.
- Activated Tgfbr complexes (RT) suppress Pax6 function via $\frac{[P]}{[RT].[P]+1}$.
- Tgfb2 is produced as a dimer, hence $\left(a_T . \frac{[P]}{[RT].[P]+1} \right)^2$.
- Tgfb2 is sequestered in the formation of Fst:Tgfb2 complexes (FT) via $-a_{FT} . [F]^2 . [T]$, and released again via $+b_{FT} . [FT]$. This implements the two-way reaction: $1 \, T + 2 \, F \underset{b_{FT}}{\overset{a_{FT}}{\rightleftharpoons}} 1 \, FT$, whereby one Tgfb2 dimer reversibly associates with two Fst monomers.
- Tgfb2 is irreversibly consumed in the formation of activated receptor complexes (RT) via $-a_{RT} . [R]^4 . [T]$. This implements the one-way reaction: $1 \, T + 4 \, R \overset{a_{RT}}{\longrightarrow} 1 \, RT$, whereby one Tgfb2 dimer associates with four Tgfbr monomers (two type I receptors + two type II receptors). This reaction is irreverible since cells internalise and ultimately turn over activated receptor complexes.
- Tgfb2 is turned over via $-b_T . [T]$.
- Tgfb2 diffuses via $D_T \frac{d^2[T]}{dx^2}$.

Fst:

$$\frac{\partial [F]}{\partial t} = a_F . \frac{[P]}{[RT].[P]+1} - 2.a_{FT} . [F]^2 . [T] + 2.b_{FT} . [FT] - b_F . [F] + D_F \frac{d^2[F]}{dx^2}$$

Where,

- Pax6 (P) positively regulates Fst expression via Michaelis-Menton kinetics $\frac{[P]}{[P]+1}$.
- Activated Tgfbr complexes (RT) suppress Pax6 function via $\frac{[P]}{[RT].[P]+1}$.
- Pairs of Fst monomers are consumed in the formation of Fst:Tgfb2 complexes (FT) via $-2.a_{FT} . [F]^2 . [T]$, and released again via $+2.b_{FT} . [FT]$. This implements the two-way reaction: $1 \, T + 2 \, F \underset{b_{FT}}{\overset{a_{FT}}{\rightleftharpoons}} 1 \, FT$, whereby one Tgfb2 dimer reversibly associates with two Fst monomers.
- Fst is turned over via $-b_F . [F]$.
- Fst diffuses via $D_F \frac{d^2[F]}{dx^2}$.

Fst:Tgfb2 complex:

$$\frac{\partial [FT]}{\partial t} = a_{FT} . [F]^2 . [T] - b_{FT} . [FT] + D_{FT} \frac{d^2[FT]}{dx^2}$$

Where,

- A pair of Fst monomers and a single Tgfb2 dimer associate via $a_{FT}.[F]^2.[T]$ forming a Fst:Tgfb2 complex, which dissociates via $-b_{FT}.[FT]$. These terms implement the two-way reaction: $1\,\mathrm{T} + 2\,\mathrm{F} \underset{b_{FT}}{\overset{a_{FT}}{\rightleftharpoons}} 1\,\mathrm{FT}$.

- Fst:Tgfb2 diffuses via $D_{FT}\dfrac{d^2[FT]}{dx^2}$.

Activated Tgfbr complex:

$$\frac{\partial[RT]}{\partial t} = a_{RT}.[R]^4.[T] - b_{RT}.[RT]$$

R implementation:

The function 'modelB' implements the above system of equations and calculates the rates of change in concentration for each molecular species. The code breaks down into the following steps:

- Fetch the current system state (i.e. molecular concentrations for each 'cell' in the tissue).

- Evaluate the reaction terms for each molecular species.

- Evaluate diffusion terms for diffusible species. Diffusion terms are written for zero-flux boundary conditions. Terms for periodic boundaries are provided but commented out due to considerably longer execution times.

- Combine reaction and diffusion terms to determine rates of change for each species.

- Update a progress bar.

- Return the list of rates.

In [6]:
```r
library(deSolve)
library(ReacTran)
library(viridis)
library(astro)
library(ggplot2)
library(readbitmap)

## ================
## Model equations
## ================
modelB <- function(time, state, parms) {
  with (as.list(parms), {

    # Generalise this: state[ ( (i-1)*N.B + 1) : (i * N.B) ]
    PAX6  <- state[1:N.B]
    TGFB2 <- state[(N.B+1):(2*N.B)]
    FST   <- state[(2*N.B+1):(3*N.B)]
    FT    <- state[(3*N.B+1):(4*N.B)]
    TGFBR <- state[(4*N.B+1):(5*N.B)]
    RT    <- state[(5*N.B+1):(6*N.B)]

    ## Reaction terms:
    reacPAX6  <- (
                  + (aPAX6 * PAX6) / (RT * PAX6 + 1)
                  - (bPAX6 * PAX6)
                  )
    reacTGFB2 <- (
                  + (aTGFB2 * PAX6 / ( RT * PAX6 + 1 ) ) ^ 2
                  - (aFT * FST * FST * TGFB2 )
                  + (bFT * FT)
                  - (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                  - (bTGFB2 * TGFB2)
                  )
    reacFST   <- (
                  + (aFST * PAX6) / (RT * PAX6 + 1)
                  - (2 * aFT * FST * FST * TGFB2)
                  + (2 * bFT * FT)
                  - (bFST * FST)
                  )
    reacFT    <- (
                  + (aFT * FST * FST * TGFB2)
                  - (bFT * FT)
                  )
    reacTGFBR <- rep(0, N.B) # Concentration does not vary throughout simulati
on
    reacRT    <- (
                  + (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                  - (bRT * RT)
                  )

    ## Diffusion terms - periodic boundary
    ## NOTE: these take longer to compute
    #diffTGFB2 <- tran.1D(C = TGFB2,
    #                     C.up = TGFB2[N.B],
    #                     C.down = TGFB2[1],
    #                     D = dTGFB2,
    #                     dx = xgridA)$dC
    #diffFST   <- tran.1D(C = FST,
    #                     C.up = FST[N.B],
    #                     C.down = FST[1],
    #                     D = dFST,
    #                     dx = xgridA)$dC
    #diffFT    <- tran.1D(C = FT,
    #                     C.up = FT[N.B],
    #                     C.down = FT[1],
```

```
#                       D = dFT,
#                       dx = xgridA)$dC

## Diffusion terms - zero-flux boundary
diffTGFB2 <- tran.1D(C = TGFB2,
                     C.up = TGFB2[1],
                     C.down = TGFB2[N.B],
                     D = dTGFB2,
                     dx = xgridB)$dC
diffFST   <- tran.1D(C = FST,
                     C.up = FST[1],
                     C.down = FST[N.B],
                     D = dFST,
                     dx = xgridB)$dC
diffFT    <- tran.1D(C = FT,
                     C.up = FT[1],
                     C.down = FT[N.B],
                     D = dFT,
                     dx = xgridB)$dC

## Diffusion terms - fixed boundary
#diffTGFB2 <- tran.1D(C = TGFB2,
#                     C.up = 0,
#                     C.down = 0,
#                     D = dTGFB2,
#                     dx = xgridA)$dC
#diffFST   <- tran.1D(C = FST,
#                     C.up = 0,
#                     C.down = 0,
#                     D = dFST,
#                     dx = xgridA)$dC
#diffFT    <- tran.1D(C = FT,
#                     C.up = 0,
#                     C.down = 0,
#                     D = dFT,
#                     dx = xgridA)$dC

## Rates of change = reaction  + diffusion:
deltaPAX6          = reacPAX6
deltaTGFB2         = reacTGFB2 + diffTGFB2
deltaFST           = reacFST   + diffFST
deltaFT            = reacFT    + diffFT
deltaTGFBR         = reacTGFBR
deltaRT            = reacRT

setTxtProgressBar(pb, time)

return (list(c(deltaPAX6, deltaTGFB2, deltaFST, deltaFT, deltaTGFBR, delta
RT)))
  })
}
```

Parameters:

In [7]:
```r
## ===================
## Model parameters:
## ===================

## Space:
L.B      <- 50                                   # Length of the 1D tissue
N.B      <- 100                                  # Number of 1D 'cells'
dxB      <- L.B/N.B                              # Length of one 1D 'cell'
xgridB <- setup.grid.1D(N = N.B, x.down = L.B)   # Output wanted at these cell
 positions

## Time:
TimeB    <- 5000                        # Duration of the simulation
StepsB   <- 40                          # Number of time-steps
dtB      <- TimeB/StepsB                # Size of each time-step
timesB   <- seq(0, TimeB, by = dtB)     # Output wanted at these time intervals

## Molecuar Species
s.names = c("PAX6",
            "TGFB2 Dimer",
            "FST Monomer",
            "FST:TGFB2\nHeterotetramer",
            "TGFBR Monomer",
            "TGFB2:TGFBR\nComplex"
           )
SPECIES = length(s.names)

# Setup text progress bar - called by model function
pbB <- txtProgressBar(min = 0, max = TimeB, style = 3)

## Rate constants:
parmsB <- c(pb      <- pbB,     # Progress bar
            xgrid   <- xgridB,

            aPAX6   = 0.3,      # Production rate constant for PAX6
            bPAX6   = 0.1,      # Decay rate constant for PAX6

            aTGFB2 = 0.25,      # Production rate constant for TGFB2 dimer
            bTGFB2 = 0.1,       # Deacy rate constant for TGFB2 dimer
            dTGFB2 = 1,         # Diffusion rate constant for TGFB2 dimer

            aFST   = 1.5,       # Production rate constant for FST
            bFST   = 0.1,       # Decay rate constant for FST
            dFST   = 1,         # Diffusion rate constant for FST

            aFT    = 5,         # Association rate constant for FST:TGFB2 comple
x
            bFT    = 1,         # Dissociation rate constant for FST:TGFB2 compl
ex
            dFT    = 50,        # Diffusion rate constant for FST:TGBF2 complex

            aRT    = 20,        # Association rate constant for TGFBR:TGFB2 comp
lex
            bRT    = 0.1        # Decay rate constant for TGFBR:TGFB2 complex
           )
```

| |

0%

Initial Conditions:

The initial conditions describe a situation in which Pax6 expression is restricted to one region within the simulated tissue, corresponding to the distal optic vesicle. This is analagous to the situation in the embryo, where BMPs from the presumptive lens ectoderm promote Pax6 expression in the distal optic vesicle (see main text). Initial distributions of all other molecular species are uniform.

```
In [8]:   ## ==================
          ## Initial conditions:
          ## ==================
          stateB <- c(jitter( rep(1, N.B), 0.01, 0.01), #PAX6 - no pre-pattern
                      rep(0, N.B),                       # TGFB2
                      rep(0, N.B),                       # FST
                      rep(0, N.B),                       # FT
                      rep(1, N.B),                       # TGFBR
                      rep(0, N.B)                        # RT
                     )
```

Run Simulation:

```
In [9]:   ## ==================
          # Run the simulation:
          ## ==================
          outB <- ode.1D(y        = stateB,
                         times    = timesB,
                         func     = modelB,
                         parms    = parmsB,
                         maxsteps = 10000000,
                         nspec    = SPECIES,
                         names    = s.names
                        )

          # Close the progress bar
          close(pb)
```

```
  |==========================================================================| 10
0%
```

In [10]:
```r
## ============================
## Plot the output via ggplot2
## ============================
library(ggplot2)
print("Formatting simulation data...")
# Create data.frame for display data
x <- seq(from=dxB,to=N.B*dxB, by=dxB) # Width of tissue
t <- timesB                           # Time steps
s <- s.names                          # Species names
dataB <- expand.grid(X=x, Species=s, Time=t, Concentration= 0.0)

# Read simulation output into data.frame ready for display
# For each species...
for (s in seq(from=1,to=SPECIES,by=1)) {
    # And working backwards for each timepoint...
    for (t in timesB) {
        # Find column range in output for current species (timepoints are in s
eperate rows)
        s.start <- 2 + N.B * ( s - 1 )
        s.end   <- s.start + N.B - 1
        # Extract data for current timepoint (row) and species (columns)
        out.ACT <- outB[1+t/dtB, s.start:s.end]
        # Normalise for this species at this timepoint only
        #out.ACT <- out.ACT / max(out.ACT)
        # Find rows in data.frame for current species and timepoint
        s.index <- dataB$Species == s.names[s] # Rows for species
        t.index <- dataB$Time    == t          # Rows for timeopoint
        # Insert data for current species and timepoint
        dataB[s.index & t.index, 4] <- out.ACT
    }
    # Normalise across all timepoints for each species
    maximum <- max(dataB[s.index, 4])
    dataB[s.index, 4] <- dataB[s.index, 4] / maximum
}
print("...done!")

# Generate multi-faceted plot for endpoint...
print("Rendering endpoint plot...")
just.PAX6 <- dataB$Species == "PAX6"
just.RT   <- dataB$Species == s.names[6]
ggplot(dataB, aes(x=Time, y=X, z=Concentration)) +
geom_tile(aes(fill = Concentration ) ) +
facet_wrap(~ Species, ncol=3) +
theme_bw() +
labs(fill = "Normalised\nConcentration") +
ylab("Anterior-Posterior Axis (AU)") +
xlab("Time (AU)") +
theme(text = element_text(size=12) ) +
scale_fill_viridis(option="B")
print("...done!")
```

```
[1] "Formatting simulation data..."
[1] "...done!"
[1] "Rendering endpoint plot..."
[1] "...done!"
```
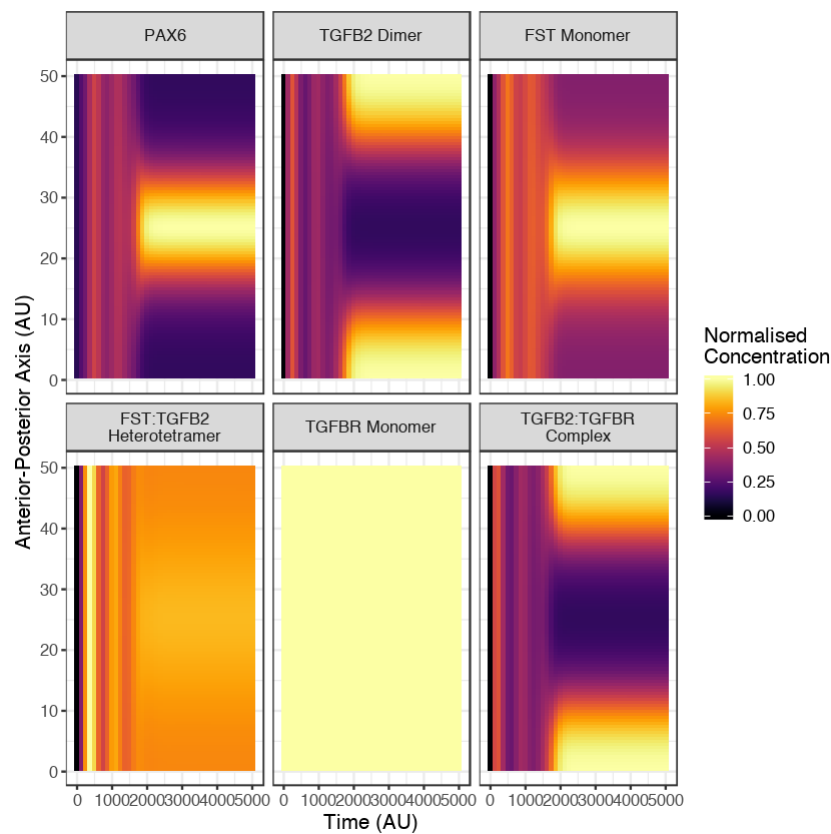

Figure S4. 1-D numerical simulation of Model B showing the normalised concentration distributions of each molecular species evolving over time. Brighter colours = higher concentrations.

Model C: 2D simulation of Reaction-Diffusion Model B:

This simulation is intended to mimic an optic vesicle explant cultured *in vitro*. The model implementation is re-wrtitten as function "modelC". The reaction terms are identical to Model B, but diffusion terms are re-written to facilitate diffusion in two dimensions within a domain of arbitrary shape. This arbitrary domain shape is loaded from a png image file within the Parameters setion. Diffusion occurs only within this domain (the diffusion multipliers DIFF.X and DIFF.Y equal 1 inside this domain and 0 outside), hence the boundary conditions are zero-flux.

In [11]:
```r
library(deSolve)
library(ReacTran)
library(gganimate)
library(ggplot2)
library(gifski)
library(png)
library(viridis)
library(av)
library(raster)
library(readbitmap)

## ================
## Model equations
## ================
modelC <- function(time, state, parms) {
  with (as.list(parms), {
    # Generalise this: state[ ( (i-1)*N + 1) : (i * N) ]
    PAX6  <- DOMAIN * state[1:N]
    TGFB2 <- DOMAIN * state[(N+1):(2*N)]
    FST   <- DOMAIN * state[(2*N+1):(3*N)]
    FT    <- DOMAIN * state[(3*N+1):(4*N)]
    TGFBR <- DOMAIN * state[(4*N+1):(5*N)]
    RT    <- DOMAIN * state[(5*N+1):(6*N)]

    ## Reaction terms:
    reacPAX6  <- (
                  + (aPAX6 * PAX6) / (RT * PAX6 + 1)
                  - (bPAX6 * PAX6)
                )
    reacTGFB2 <- (
                  + (aTGFB2 * PAX6 / ( RT * PAX6 + 1 ) ) ^ 2
                  - (aFT * FST * FST * TGFB2 )
                  + (bFT * FT)
                  - (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                  - (bTGFB2 * TGFB2)
                )
    reacFST   <- (
                  + (aFST * PAX6) / (RT * PAX6 + 1)
                  - (2 * aFT * FST * FST * TGFB2)
                  + (2 * bFT * FT)
                  - (bFST * FST)
                )
    reacFT    <- (
                  + (aFT * FST * FST * TGFB2)
                  - (bFT * FT)
                )
    reacTGFBR <- rep(0, N.B) # Concentration does not vary throughout simulati
on
    reacRT    <- (
                  + (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                  - (bRT * RT)
                )

    ## Find boundaries:
    TGFB2.matrix <- matrix(data = TGFB2, nrow = X.N, ncol = Y.N)
    TGFB2.y.up   <- TGFB2.matrix[    , 1  ]
    TGFB2.y.down <- TGFB2.matrix[    , Y.N]
    TGFB2.x.up   <- TGFB2.matrix[1   ,    ]
    TGFB2.x.down <- TGFB2.matrix[X.N,     ]
    FST.matrix <- matrix(data = FST, nrow = X.N, ncol = Y.N)
    FST.y.up   <- FST.matrix[    , 1  ]
    FST.y.down <- FST.matrix[    , Y.N]
    FST.x.up   <- FST.matrix[1   ,    ]
    FST.x.down <- FST.matrix[X.N,     ]
    FT.matrix <- matrix(data = FT, nrow = X.N, ncol = Y.N)
```

```
      FT.y.up   <- FT.matrix[    , 1  ]
      FT.y.down <- FT.matrix[    , Y.N]
      FT.x.up   <- FT.matrix[1  ,    ]
      FT.x.down <- FT.matrix[X.N,    ]

      ### NO DIFFUSION AT DOMAIN BOUNDARIES = ZERO-FLUX BOUNDARY
      diffTGFB2 <- tran.2D(C = TGFB2.matrix, D.x = dTGFB2*DIFF.X, D.y = dTGFB2*D
IFF.Y, grid = grid_2d,
                            C.x.up = TGFB2.x.up, C.x.down = TGFB2.x.down,
                            C.y.up = TGFB2.y.up, C.y.down = TGFB2.y.down)$dC
      diffFST   <- tran.2D(C = FST.matrix,   D.x = dFST*DIFF.X,   D.y = dFST*DIF
F.Y,   grid = grid_2d,
                            C.x.up = FST.x.up,   C.x.down = FST.x.down,
                            C.y.up = FST.y.up,   C.y.down = FST.y.down  )$dC
      diffFT    <- tran.2D(C = FT.matrix,    D.x = dFT*DIFF.X,    D.y = dFT*DIF
F.Y,    grid = grid_2d,
                            C.x.up = FT.x.up,    C.x.down = FT.x.down,
                            C.y.up = FT.y.up,    C.y.down = FT.y.down   )$dC

      ### REMOVING DIFF.Y AND DIFF.X = DIFFUSION AT BOUNDARIES = FIXED BOUNDARY
      ### RESULTS IN CONCENTRIC STRIPES AS DOMAIN SIZE INCREASES
      ### THIS IS INCONSISTENT WITH OBSERVATIONS FROM EXPLANT EXPERIMENTS
      #diffTGFB2 <- tran.2D(C = TGFB2.matrix, D.x = dTGFB2, D.y = dTGFB2, grid =
grid_2d,
      #                     C.x.up = TGFB2.x.up, C.x.down = TGFB2.x.down,
      #                     C.y.up = TGFB2.y.up, C.y.down = TGFB2.y.down)$dC
      #diffFST   <- tran.2D(C = FST.matrix,   D.x = dFST,   D.y = dFST,   grid =
grid_2d,
      #                     C.x.up = FST.x.up,   C.x.down = FST.x.down,
      #                     C.y.up = FST.y.up,   C.y.down = FST.y.down  )$dC
      #diffFT    <- tran.2D(C = FT.matrix,    D.x = dFT,    D.y = dFT,    grid =
grid_2d,
      #                     C.x.up = FT.x.up,    C.x.down = FT.x.down,
      #                     C.y.up = FT.y.up,    C.y.down = FT.y.down   )$dC

      ## Rates of change = DOMAIN * (Reaction  + Diffusion):
      deltaPAX6          = DOMAIN * (reacPAX6)
      deltaTGFB2         = DOMAIN * (reacTGFB2 + diffTGFB2)
      deltaFST           = DOMAIN * (reacFST   + diffFST)
      deltaFT            = DOMAIN * (reacFT    + diffFT)
      deltaTGFBR         = DOMAIN * (reacTGFBR)
      deltaRT            = DOMAIN * (reacRT)

      setTxtProgressBar(pb, time)

      return (list(c(deltaPAX6, deltaTGFB2, deltaFST, deltaFT, deltaTGFBR, delta
RT)))
  })
}
```

```
Warning message:
"package 'gganimate' was built under R version 3.5.2"
Warning message:
"package 'raster' was built under R version 3.5.2"
Loading required package: sp


Attaching package: 'raster'


The following object is masked from 'package:gganimate':

    animate


The following objects are masked from 'package:MASS':

    area, select
```

Parameters:

In [12]:
```r
## ===================
## Model parameters:
## ===================

## Space:

# Read bitmap of 2D space
imgC <- read.bitmap("./OV_Explant4.png")
#imgC <- read.bitmap("./OV_Explant2.png")
# Extract dimension from bitmap
X.N <- dim(imgC)[1] #Number of elements (cells) across the width
Y.N <- dim(imgC)[2] #Number of elements (cells) across the height
N   <- X.N*Y.N      # Number of elements (cellS) in total
# Convert (#hex) bitmap to vector of integers
v <- as.integer(imgC)
# Convert this vector to matrix, just to check its dimensions
m <- matrix(v, X.N, Y.N)
# Convert matrix back to vector and use this as the model domain
DOMAIN <- as.vector(m)
# Use matrix to calculate which pixel/cell borders permit diffusion in X and Y
directions
DIFF.X <- rbind(rep(0.0, Y.N), floor( (m[-1,  ] + m[-nrow(m),        ]) / 2),
rep(0.0, Y.N) )
DIFF.Y <- cbind(rep(0.0, X.N), floor( (m[  ,-1] + m[        ,-ncol(m)]) / 2),
rep(0.0, X.N) )

X.L <- 13                                            # Width  of the 2D tissue
Y.L <- 9                                             # Height of the 2D tissue
dx  <- X.L/X.N                                       # Length of one 1D elemen
t in X
dy  <- Y.L/Y.N                                       # Length of one 1D elemen
t in Y
xgrid   <- setup.grid.1D(N = X.N, x.up = 0, L = X.L) # Output wanted at these
 X positions
ygrid   <- setup.grid.1D(N = Y.N, x.up = 0, L = Y.L) # Output wanted at these
 Y positions
grid_2d <- setup.grid.2D(xgrid, ygrid)               # Output wanted at these
 grid positions

## Time:
TimeC   <- 5000                                      # Duration of the simulation
StepsC  <- 20                                        # Number of time-steps
dtC     <- TimeC/StepsC                              # Size of each time-step
timesC  <- seq(0, TimeC, by = dtC)                   # Output wanted at these time in
tervals

## Molecuar Species
s.names = c("PAX6",
            "TGFB2 Dimer",
            "FST Monomer",
            "FST:TGFB2\nHeterotetramer",
            "TGFBR Monomer",
            "TGFB2:TGFBR\nComplex")
SPECIES = length(s.names)    # Count the number of species

# Setup text progress bar - called by model function
pbC <- txtProgressBar(min = 0, max = TimeC, style = 3)

## Rate constants:
parmsC <- c(pb      <- pbC,    # Progress bar

            aPAX6  = 0.3,     # Production rate constant for PAX6
            bPAX6  = 0.1,     # Decay rate constant for PAX6

            aTGFB2 = 0.25,    # Production rate constant for TGFB2 dimer
```

```
              bTGFB2 = 0.1,        # Deacy rate constant for TGFB2 dimer
              dTGFB2 = 1,          # Diffusion rate constant for TGFB2 dimer

              aFST   = 1.5,        # Production rate constant for FST
              bFST   = 0.1,        # Decay rate constant for FST
              dFST   = 1,          # Diffusion rate constant for FST

              aFT    = 5,          # Association rate constant for FST:TGFB2 comple
x
              bFT    = 1,          # Dissociation rate constant for FST:TGFB2 compl
ex
              dFT    = 50,         # Diffusion rate constant for FST:TGBF2 complex

              aRT    = 20,         # Association rate constant for TGFBR:TGFB2 comp
lex
              bRT    = 0.1         # Decay rate constant for TGFBR:TGFB2 complex
              )
```

| |

0%

Initial Conditions:

The initial conditions describe a situation in which Pax6 expression is graded from distal-high to proximal-low.

In [13]:
```
## ====================
## Initial conditions:
## ====================
Pax6.Grad <- rep(seq(from = 3,     to = 5, length.out = Y.N), X.N)
Pax6.Grad <- as.vector( t( matrix(Pax6.Grad, ncol = X.N, nrow = Y.N) ) )

stateC <- c(DOMAIN * Pax6.Grad,  # PAX6  - graded
            DOMAIN * rep(0, N),  # TGFB2
            DOMAIN * rep(0, N),  # FST
            DOMAIN * rep(0, N),  # FT
            DOMAIN * rep(1, N),  # TGFBR
            DOMAIN * rep(0, N)   # RT
            )
```


Run Simulation:

```
In [14]:  ## ==================
          # Run the simulation:
          ## ==================
          cat("Initializing simulation...")
          outputC <- ode.2D(y       = stateC,
                            times   = timesC,
                            func    = modelC,
                            parms   = parmsC,
                            nspec   = SPECIES,
                            names   = s.names,
                            dimens  = c(X.N, Y.N),
                            method  = "lsode",
                            lrw     = 7e+08
                           )
          cat(" done!")
          flush.console()
```

```
   |=====================================================================| 10
0% done!
```

```
In [15]:  ## ==================
          ## Format the output
          ## ==================
          # Create data.frame for display data
          x <- seq(from=dx, to=X.L,  by=dx) # Width of tissue
          y <- seq(from=dy, to=Y.L,  by=dy) # Height of tissue
          t <- timesC                       # Time steps
          s <- s.names                      # Species names
          dataC <- expand.grid(X=x, Y=y, Species=s, Time=t, Concentration= 0.0)
          cat("Formatting output...")
          # Read simulation output into data.frame ready for display
          # For each species...
          for (s in seq(from=1,to=SPECIES,by=1)) {
              # And for each subsequent timepoint...
              for (t in timesC) {
                  # Find column range in output for current species (timepoints are in s
          eperate rows)
                  s.start <- 2 + N * ( s - 1 )
                  s.end   <- s.start + N - 1
                  # Extract data for current timepoint (row) and species (columns)
                  out.ACT <- outputC[1+ (t/dtC), s.start:s.end]
                  # Normalise for this species at this timepoint only
                  #out.ACT <- out.ACT / max(out.ACT)
                  # Find rows in data.frame for current species and timepoint
                  s.index <- dataC$Species == s.names[s] # Rows for species
                  t.index <- dataC$Time    == t          # Rows for timeopoint
                  # Insert data for current species and timepoint
                  dataC[s.index & t.index, 5] <- out.ACT
              }
              # Normalise across all timepoints for each species
              maximum <- max(dataC[s.index, 5])
              dataC[s.index, 5] <- dataC[s.index, 5] / maximum
          }
          cat("done!")
          flush.console()
```

```
Formatting output...done!
```

```
In [16]: ## ============================
         ## Plot the output via ggplot2
         ## ============================
         # Generate multi-faceted endpoint plot...
         cat("Rendering plot...")
         just.PAX6 <- dataC$Species == "PAX6"
         p <- ggplot(dataC, aes(X, Y, z=Concentration, frame=Time)) +
         geom_tile(aes(fill = Concentration ) ) +
         facet_wrap(~ Species, ncol=3) +
         theme_bw() +
         theme(text = element_text(size=12) ) +
         coord_fixed() +
         scale_fill_viridis(option="B") +
         ylab("Proximal-Distal Axis (AU)") +
         xlab("Anterior-Posterior Axis (AU)") +
         labs(title = "Normalised concentrations at endpoint") +
         labs(fill = "Normalised\nConcentration")

         plot(p)

         cat("done!")
         flush.console()
```
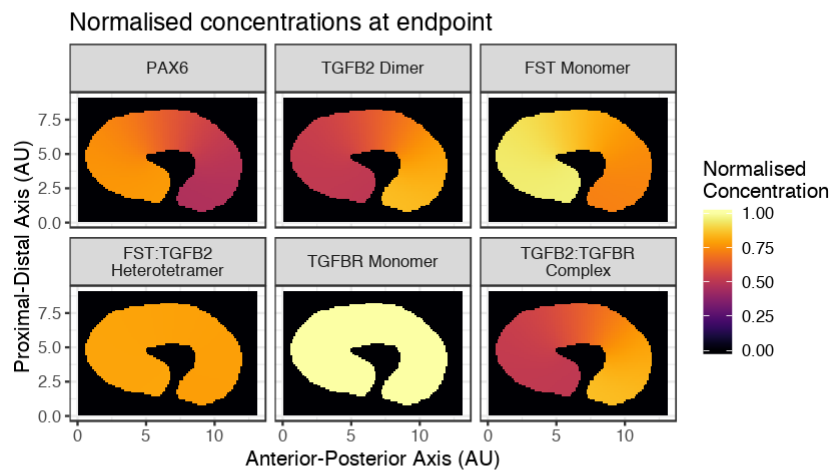
```
Rendering plot...done!
```


Figure S5. 2-D numerical simulation of Model C showing the normalised concentration distributions of each molecular species at endpoint (t = 5000). Brighter colours = higher concentrations.

In [17]:
```r
## =====================================
## Plot output for Pax6 as aimated Gif
## =====================================
# Generate multi-faceted plot, animated over timepoints...
cat("Building plot...")
just.PAX6 <- dataC$Species == "PAX6"
p <- ggplot(dataC, aes(X, Y, z=Concentration, frame= Time)) +
geom_tile(aes(fill = Concentration ) ) +
facet_wrap(~ Species, ncol=3) +
theme_bw() +
theme(text = element_text(size=12) ) +
coord_fixed() +
scale_fill_viridis(option="B") +
transition_time(Time) + # Animate over timepoints
ylab("Proximal-Distal Axis (AU)") +
xlab("Anterior-Posterior Axis (AU)") +
labs(title = "Normalised concentrations at time = {frame_time}") +
labs(fill = "Normalised\nConcentration")

cat("done!")
flush.console()

cat("\nRendering plot...")

## Render animation as an ffmpeg video file and dispay
#anC1 <- animate(p,
#                fps=10,
#                nframes=2*StepsC,
#                width=720,
#                height=480,
#                start_pause=StepsC/2,
#                end_pause=StepsC/2,
#                rewind=FALSE,
#                renderer=av_renderer() )
#contents <- base64enc::base64encode(anC1)
#tag <- '<video src="data:video/mp4;base64,%s" controls autoplay></video>'
#IRdisplay::display_html(sprintf(tag, contents))

## Save ffmpeg video to file
#anim_save(animation = anC1, filename = "Model_C.mp4")

## Render animation as a GIF file and display
suppressMessages({
    anC2 <- gganimate::animate(p,
                               fps=10,
                               nframes=2*StepsC,
                               width=720,
                               height=480,
                               start_pause=StepsC/2,
                               end_pause=StepsC/2
                               )
})
contents <- base64enc::base64encode(anC2)
tag <- '<center><img src="data:image/gif;base64,%s"></center>'
IRdisplay::display_html(sprintf(tag, contents))

cat("done!")
flush.console()

## Save GIF animation to file
#anim_save(animation = anC2, filename = "Model_C.gif")
```

```
Building plot...done!
Rendering plot...
```
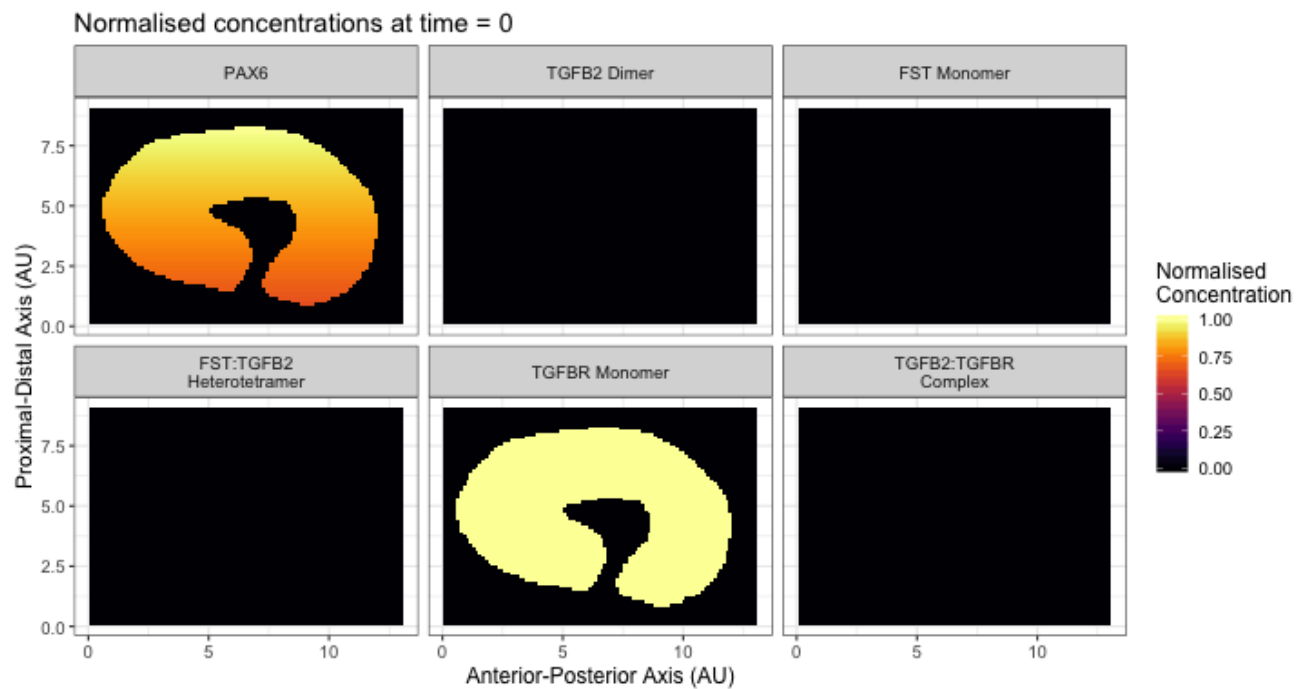


```
done!
```

Figure S6. 2-D numerical simulation of Model C showing the normalised concentration distributions of each molecular species evolving over time. In print/PDF form, this plot shows only the initial conditions for the simulation (i.e. at t = 0), while the endpoint (at t = 5000) is shown in Supp. Fig. S5. In the online Jupyter Notebook version (see methods section of paper for URL) this plot is animated. The animated plot is included as Supp. Movie 3. Brighter colours = higher concentrations.

Model D: Intrinsic SHH gradient restricts Pax6 re-polarisation to the former dorsal side

This model adds Shh as a 7th molecular species.

The governing equation for Pax6 is modified such that Pax6 expression is inhibited by Shh in addition to activated Tgfbr complexes.

Pax6:

$$\frac{\partial [P]}{\partial t} = a_P . \frac{[P]}{([RT] + [SHH]).[P] + 1} - b_P . [P]$$

Where,

- Pax6 (P) positively autoregulates via Michaelis-Menton kinetics $\frac{[P]}{[P]+1}$.
- Activated Tgfbr complexes (RT) and Shh (SHH) suppress Pax6 function via $\frac{[P]}{([RT]+[SHH]).[P]+1}$.
- Pax6 is turned over via $-b_P . [P]$.

The initial conditions are extended to include a Shh concentration gradient.

In [18]:

```r
library(deSolve)
library(ReacTran)
library(gganimate)
library(ggplot2)
library(gifski)
library(png)
library(viridis)
library(av)
library(raster)
library(readbitmap)

## ================
## Model equations
## ================
modelD <- function(time, state, parms) {
  with (as.list(parms), {
    # Generalise this: state[ ( (i-1)*N + 1) : (i * N) ]
    PAX6  <- DOMAIN * state[(0*N+1):(1*N)]
    TGFB2 <- DOMAIN * state[(1*N+1):(2*N)]
    FST   <- DOMAIN * state[(2*N+1):(3*N)]
    FT    <- DOMAIN * state[(3*N+1):(4*N)]
    TGFBR <- DOMAIN * state[(4*N+1):(5*N)]
    RT    <- DOMAIN * state[(5*N+1):(6*N)]
    SHH   <- DOMAIN * state[(6*N+1):(7*N)]

    ## Reaction terms:
    reacPAX6  <- (
                  + (aPAX6 * PAX6) / ( (RT + SHH) * PAX6 + 1)
                  - (bPAX6 * PAX6)
                 )
    reacTGFB2 <- (
                  + (aTGFB2 * PAX6 / ( RT * PAX6 + 1 ) ) ^ 2
                  - (aFT * FST * FST * TGFB2 )
                  + (bFT * FT)
                  - (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                  - (bTGFB2 * TGFB2)
                 )
    reacFST   <- (
                  + (aFST * PAX6) / (RT * PAX6 + 1)
                  - (2 * aFT * FST * FST * TGFB2)
                  + (2 * bFT * FT)
                  - (bFST * FST)
                 )
    reacFT    <- (
                  + (aFT * FST * FST * TGFB2)
                  - (bFT * FT)
                 )
    reacTGFBR <- rep(0, N.B) # Concentration does not vary throughout simulati
on

    ### For simulating Tgfb LOF, the production term may be multiplied by zero
###
    reacRT    <- (
                  + (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2)
                  #+ (aRT * TGFBR * TGFBR * TGFBR * TGFBR * TGFB2) * 0
                  - (bRT * RT)
                 )
    reacSHH   <- rep(0, N) # Concentration does not vary throughout simulation

    ## Find boundaries:
    TGFB2.matrix <- matrix(data = TGFB2, nrow = X.N, ncol = Y.N)
    TGFB2.y.up   <- TGFB2.matrix[   , 1  ]
    TGFB2.y.down <- TGFB2.matrix[   , Y.N]
    TGFB2.x.up   <- TGFB2.matrix[1  ,    ]
    TGFB2.x.down <- TGFB2.matrix[X.N,    ]
```

```
    FST.matrix <- matrix(data = FST, nrow = X.N, ncol = Y.N)
    FST.y.up   <- FST.matrix[    , 1  ]
    FST.y.down <- FST.matrix[    , Y.N]
    FST.x.up   <- FST.matrix[1  ,    ]
    FST.x.down <- FST.matrix[X.N,    ]
    FT.matrix <- matrix(data = FT, nrow = X.N, ncol = Y.N)
    FT.y.up   <- FT.matrix[    , 1  ]
    FT.y.down <- FT.matrix[    , Y.N]
    FT.x.up   <- FT.matrix[1  ,    ]
    FT.x.down <- FT.matrix[X.N,    ]


    ### NO DIFFUSION AT DOMAIN BOUNDARIES = ZERO-FLUX BOUNDARY
    diffTGFB2 <- tran.2D(C = TGFB2.matrix, D.x = dTGFB2*DIFF.X, D.y = dTGFB2*D
IFF.Y, grid = grid_2d,
                         C.x.up = TGFB2.x.up, C.x.down = TGFB2.x.down,
                         C.y.up = TGFB2.y.up, C.y.down = TGFB2.y.down)$dC
    diffFST   <- tran.2D(C = FST.matrix,   D.x = dFST*DIFF.X,   D.y = dFST*DIF
F.Y,   grid = grid_2d,
                         C.x.up = FST.x.up,   C.x.down = FST.x.down,
                         C.y.up = FST.y.up,   C.y.down = FST.y.down   )$dC
    diffFT    <- tran.2D(C = FT.matrix,    D.x = dFT*DIFF.X,    D.y = dFT*DIF
F.Y,    grid = grid_2d,
                         C.x.up = FT.x.up,    C.x.down = FT.x.down,
                         C.y.up = FT.y.up,    C.y.down = FT.y.down    )$dC


    ### REMOVING DIFF.Y AND DIFF.X = DIFFUSION AT BOUNDARIES = FIXED BOUNDARY
    ### RESULTS IN CONCENTRIC STRIPES AS DOMAIN SIZE INCREASES
    ### THIS IS INCONSISTENT WITH OBSERVATIONS FROM EXPLANT EXPERIMENTS
    #diffTGFB2 <- tran.2D(C = TGFB2.matrix, D.x = dTGFB2, D.y = dTGFB2, grid =
grid_2d,
    #                     C.x.up = TGFB2.x.up, C.x.down = TGFB2.x.down,
    #                     C.y.up = TGFB2.y.up, C.y.down = TGFB2.y.down)$dC
    #diffFST   <- tran.2D(C = FST.matrix,   D.x = dFST,   D.y = dFST,   grid =
grid_2d,
    #                     C.x.up = FST.x.up,   C.x.down = FST.x.down,
    #                     C.y.up = FST.y.up,   C.y.down = FST.y.down   )$dC
    #diffFT    <- tran.2D(C = FT.matrix,    D.x = dFT,    D.y = dFT,    grid =
grid_2d,
    #                     C.x.up = FT.x.up,    C.x.down = FT.x.down,
    #                     C.y.up = FT.y.up,    C.y.down = FT.y.down    )$dC


    ## Rates of change = DOMAIN * (Reaction  + Diffusion):
    deltaPAX6          = DOMAIN * (reacPAX6)
    deltaTGFB2         = DOMAIN * (reacTGFB2 + diffTGFB2)
    deltaFST           = DOMAIN * (reacFST   + diffFST)
    deltaFT            = DOMAIN * (reacFT    + diffFT)
    deltaTGFBR         = DOMAIN * (reacTGFBR)
    deltaRT            = DOMAIN * (reacRT)
    deltaSHH           = DOMAIN * (reacSHH)


    setTxtProgressBar(pb, time)


    return (list(c(deltaPAX6, deltaTGFB2, deltaFST, deltaFT, deltaTGFBR, delta
RT, deltaSHH)))
  })
}
```

In [19]:
```r
## ==================
## Model parameters:
## ==================

### Space:

# Read bitmap of 2D space
#imgD <- read.bitmap("./OV_Explant4.png")
imgD <- read.bitmap("./OV_Explant2.png")
# Extract dimension from bitmap
X.N <- dim(imgD)[1] #Number of elements (cells) across the width
Y.N <- dim(imgD)[2] #Number of elements (cells) across the height
N   <- X.N*Y.N      # Number of elements (cellS) in total
# Convert (#hex) bitmap to vector of integers
v <- as.integer(imgD)
## Convert this vector to matrix, just to check its dimensions
m <- matrix(v, X.N, Y.N)
# Convert matrix back to vector and use this as the model domain
DOMAIN <- as.vector(m)
# Use matrix to calculate which pixel/cell borders permit diffusion in X and Y
directions
DIFF.X <- rbind(rep(0.0, Y.N), floor( (m[-1,  ] + m[-nrow(m),        ]) / 2),
rep(0.0, Y.N) )
DIFF.Y <- cbind(rep(0.0, X.N), floor( (m[  ,-1] + m[          ,-ncol(m)]) / 2),
rep(0.0, X.N) )

X.L <- 13                                          # Width  of the 2D tissue
Y.L <- 9                                           # Height of the 2D tissue
dx  <- X.L/X.N                                     # Length of one 1D elemen
t (cell) in X
dy  <- Y.L/Y.N                                     # Length of one 1D elemen
t (cell) in Y
xgrid   <- setup.grid.1D(N = X.N, x.up = 0, L = X.L) # Output wanted at these
 X positions
ygrid   <- setup.grid.1D(N = Y.N, x.up = 0, L = Y.L) # Output wanted at these
 Y positions
grid_2d <- setup.grid.2D(xgrid, ygrid)

## Time:
TimeD   <- 5000                                    # Duration of the simulation
StepsD  <- 20                                      # Number of time-steps
dtD     <-  TimeC/StepsC                           # Size of each time-step
timesD  <- seq(0, TimeC, by = dtD)                 # Output wanted at these time in
tervals

# Molecuar Species
s.names_D = c("PAX6",
              "TGFB2 Dimer",
              "FST Monomer",
              "FST:TGFB2\nHeterotetramer",
              "TGFBR Monomer",
              "TGFB2:TGFBR\nComplex",
              "SHH")
SPECIES_D = length(s.names_D)

# Setup text progress bar - called by model function
pbD <- txtProgressBar(min = 0, max = TimeC, style = 3)

# Rate constants:
parmsD <- c(pb      <- pbD,    # Progress bar

            aPAX6  = 0.3,     # Production rate constant for PAX6
            bPAX6  = 0.1,     # Decay rate constant for PAX6

            aTGFB2 = 0.25,    # Production rate constant for TGFB2 dimer
```

```
            bTGFB2 = 0.1,        # Deacy rate constant for TGFB2 dimer
            dTGFB2 = 1,          # Diffusion rate constant for TGFB2 dimer

            aFST   = 1.5,        # Production rate constant for FST
            bFST   = 0.1,        # Decay rate constant for FST
            dFST   = 1,          # Diffusion rate constant for FST

            aFT    = 5,          # Association rate constant for FST:TGFB2 comple
x
            bFT    = 1,          # Dissociation rate constant for FST:TGFB2 compl
ex
            dFT    = 50,         # Diffusion rate constant for FST:TGBF2 complex

            aRT    = 20,         # Association rate constant for TGFBR:TGFB2 comp
lex
            bRT    = 0.1         # Decay rate constant for TGFBR:TGFB2 complex
          )
```

```
 |                                                                              |
0%
```

```
In [26]:   ## ===================
           ## Initial conditions:
           ## ===================

           # Calculate the SHH gradient:
           ### For simulating Shh loss-of-function (LOF), SHH.X may be multiplied by zero
           ###
           SHH.X <- 100 * dnorm(seq(1, X.N, by = 1), mean = X.N, sd = ( (2*X.N) / (X.L/2)
           ) )
           #SHH.X <- 100 * dnorm(seq(1, X.N, by = 1), mean = 1,   sd = ( (2*X.N) / (X.L/
           2) ) ) # Flip the gradient
           #SHH.X <- SHH.X * 0
           plot(SHH.X,
                xlab="Dorsal-Ventral Gradient",
                ylab="Shh Concentration",
                main="Shh Positional Information Gradient")

           ## ===================
           ## Initial conditions:
           ## ===================
           Pax6.Grad <- rep(seq(from = 3,     to = 5, length.out = Y.N), X.N)
           Pax6.Grad <- as.vector( t( matrix(Pax6.Grad, ncol = X.N, nrow = Y.N) ) )

           stateD <- c(DOMAIN * Pax6.Grad,          # PAX6  - graded
                       DOMAIN * rep(0, N),          # TGFB2
                       DOMAIN * rep(0, N),          # FST
                       DOMAIN * rep(0, N),          # FT
                       DOMAIN * rep(1, N),          # TGFBR
                       DOMAIN * rep(0, N),          # RT
                       DOMAIN * rep(SHH.X, Y.N)     # SHH
                      )
```
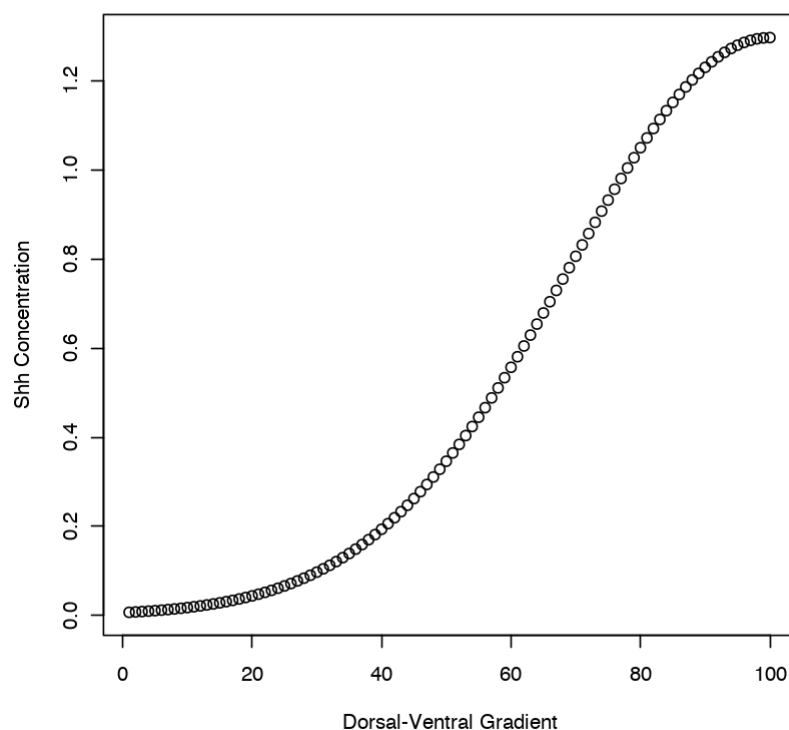

Figure S7. 1-D plot of the SHH positional information gradient used in Model D.

In [27]:
```r
## ==================
# Run the simulation:
## ==================
cat("Initializing simulation...")
outputD <- ode.2D(y       = stateD,
                  times  = timesD,
                  func   = modelD,
                  parms  = parmsD,
                  nspec  = SPECIES_D,
                  names  = s.names_D,
                  dimens = c(X.N, Y.N),
                  method = "lsode",
                  lrw    = 7e+08)
cat(" done!")
flush.console()
```

```
|============================================================== |   9
5% done!
```

In [28]:
```r
## ==================
## Format the output
## ==================
# Create data.frame for display data
x <- seq(from=dx, to=X.L,  by=dx) # Width of tissue
y <- seq(from=dy, to=Y.L,  by=dy) # Height of tissue
t <- timesD                        # Number of time steps
s <- s.names_D                     # Number of species
dataD <- expand.grid(X=x, Y=y, Species=s, Time=t, Concentration= 0.0)
cat("Formatting output...")
# Read simulation output into data.frame ready for display
# For each species...
for (s in seq(from=1,to=SPECIES_D,by=1)) {
    # And for each subsequent timepoint...
    for (t in timesD) {
        # Find column range in output for current species (timepoints are in s
eperate rows)
        s.start <- 2 + N * ( s - 1 )
        s.end   <- s.start + N - 1
        # Extract data for current timepoint (row) and species (columns)
        out.ACT <- outputD[1+ (t/dtD), s.start:s.end]
        # Normalise for this species at this timepoint only
        #out.ACT <- out.ACT / max(out.ACT)
        # Find rows in data.frame for current species and timepoint
        s.index <- dataD$Species == s.names_D[s]  # Rows for species
        t.index <- dataD$Time    == t#*dt         # Rows for timeopoint
        # Insert data for current species and timepoint
        dataD[s.index & t.index, 5] <- out.ACT
    }
    # Normalise across all timepoints for each species
    maximum <- max(dataD[s.index, 5])
    dataD[s.index, 5] <- dataD[s.index, 5] / maximum
}
cat("done!")
flush.console()
```

```
Formatting output...done!
```

```
In [29]:  ## ==================
          ## Plot the output
          ## ==================
          # Generate multi-faceted endpoint plot...
          cat("Rendering plot...")
          ggplot(dataD, aes(X, Y, z=Concentration, frame=Time)) +
          geom_tile(aes(fill = Concentration ) ) +
          facet_wrap(~ Species, ncol=3) +
          theme_bw() +
          theme(text = element_text(size=8) ) +
          coord_fixed() +
          scale_fill_viridis(option="B") +
          ylab("Proximal-Distal Axis (AU)") +
          xlab("Dorsal-Ventral Axis (AU)") +
          labs(title = "Normalised concentrations at endpoint") +
          labs(fill = "Normalised\nConcentration")

          cat("done!")
          flush.console()
```
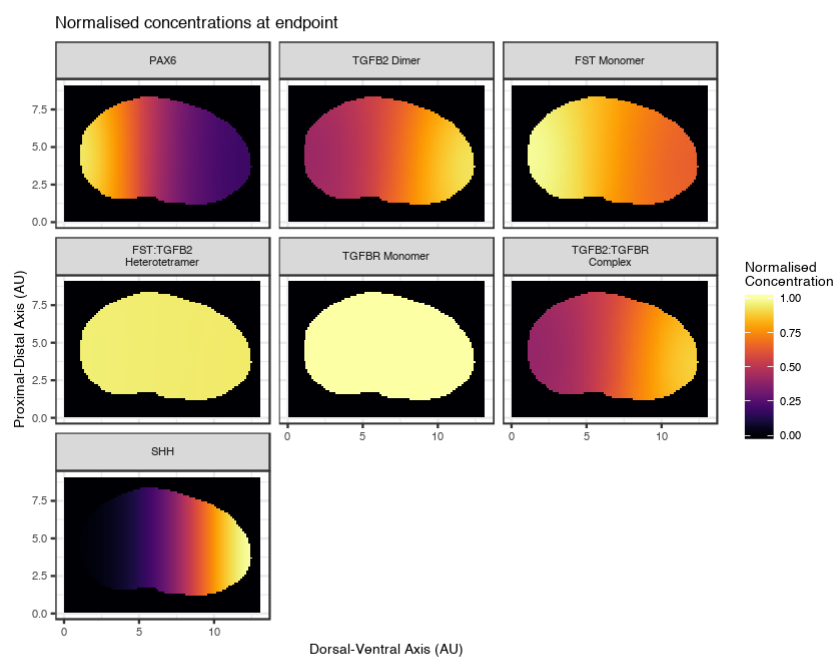
Rendering plot...done!

Figure S8. 2-D numerical simulation of Model D showing the normalised concentration distributions of each molecular species at endpoint (t = 5000). Brighter colours = higher concentrations.

In [30]:
```r
## ==================
## Plot the output
## ==================
# Generate multi-faceted plot, animated over timepoints...
cat("Building plot...")
p <- ggplot(dataD, aes(X, Y, z=Concentration, frame= Time)) +
geom_tile(aes(fill = Concentration ) ) +
facet_wrap(~ Species, ncol=3) +
theme_bw() +
theme(text = element_text(size=12) ) +
coord_fixed() +
scale_fill_viridis(option="B") +
transition_time(Time) + # Animate over timepoints
ylab("Proximal-Distal Axis (AU)") +
xlab("Dorsal-Ventral Axis (AU)") +
labs(title = "Normalised concentrations at time = {frame_time}") +
labs(fill = "Normalised\nConcentration")

cat("done!")
flush.console()

cat("\nRendering plot...")

## Render animation as an ffmpeg video file and dispay
#anD1 <- animate(p,
#                fps=10,
#                nframes=2*StepsD,
#                width=720,
#                height=480,
#                start_pause=StepsD/2,
#                end_pause=StepsD/2,
#                rewind=FALSE,
#                renderer=av_renderer() )
#contents <- base64enc::base64encode(anD1)
#tag <- '<video src="data:video/mp4;base64,%s" controls autoplay></video>'
#IRdisplay::display_html(sprintf(tag, contents))

## Save ffmpeg video to file
#anim_save(animation = anD1, filename = "Model_D.mp4")

## Render animation as a GIF file and display
suppressMessages({
    anD2 <- gganimate::animate(p,
                        fps=10,
                        nframes=2*StepsD,
                        width=720,
                        height=480,
                        start_pause=StepsD/2,
                        end_pause=StepsD/2)
})
contents <- base64enc::base64encode(anD2)
tag <- '<center><img src="data:image/gif;base64,%s"></center>'
IRdisplay::display_html(sprintf(tag, contents))

cat("done!")
flush.console()

## Save GIF animation to file
#anim_save(animation = anD2, filename = "Model_D.gif")
```
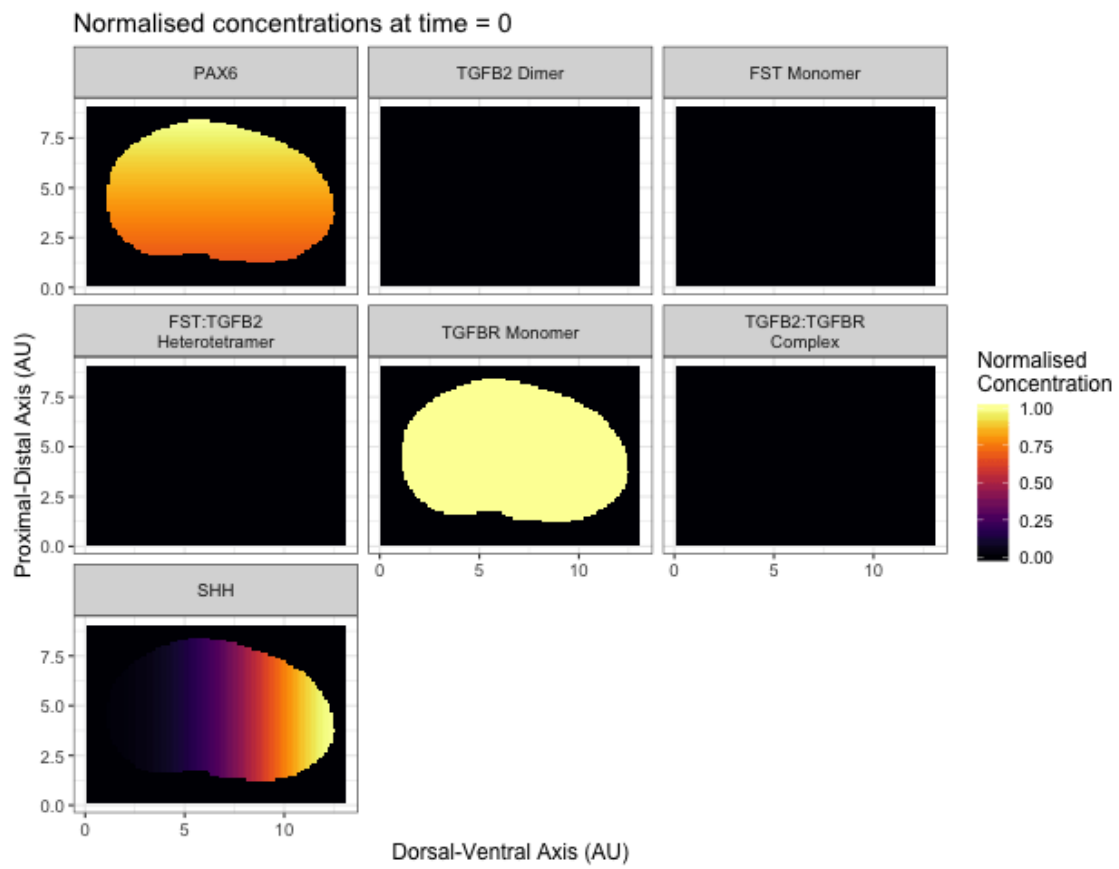
```
Building plot...done!
Rendering plot...
```


Normalised concentrations at time = 0

```
done!
```

Figure S9. 2-D numerical simulation of Model D showing the normalised concentration distributions of each molecular species evolving over time. In print/PDF form, this plot shows only the initial conditions for the simulation (i.e. at t = 0), while the endpoint (at t = 5000) is shown in Supp. Fig. S8. In the online Jupyter Notebook version (see methods section of paper for URL) this plot is animated. Brighter colours = higher concentrations.

```
In [ ]:
```
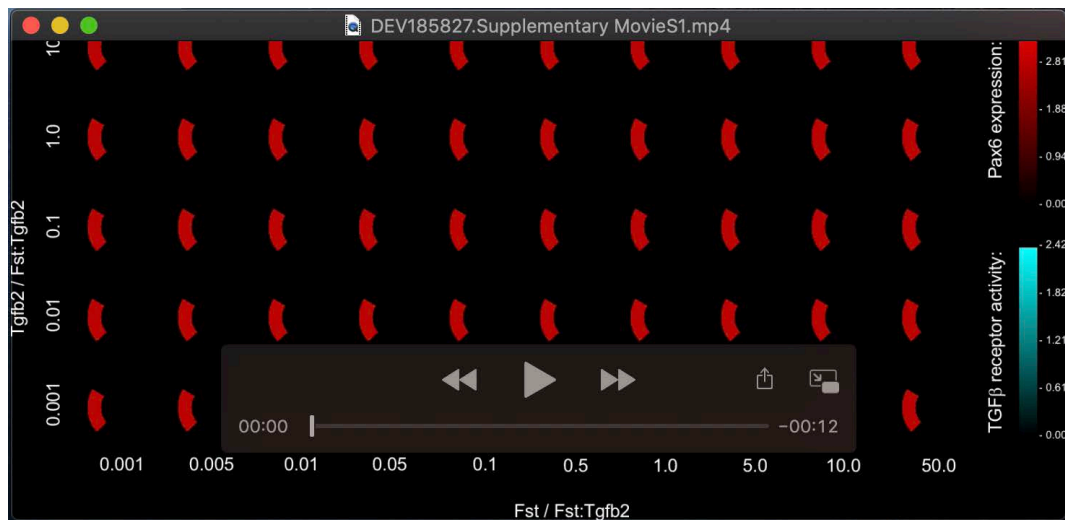
Movie 1. 1-D numerical simulations of Model A. Simulations demonstrate that local inhibition and lateral-activation of Tgfb2 signalling (i.e. Tgfb2:Tgfbr complex formation) may occur if Fst:Tgfb2 hetero-tetramer diffusion exceeds that of Fst monomers. Pax6 'expression' was restricted to just the left side of a circular tissue throughout. Simulations were performed using different relative diffusion rates for Fst monomers (x-axis) and Tgfb2 dimers (y-axis), while Fst:Tgfb2 hetero-tetramer diffusion rate was held constant. Simulations are shown three times; first with both Pax6 (red) and activated Tgfb receptor (i.e. Tgfb2:Tgfbr complex; cyan) concentrations together; second with just activated receptor concentrations; third with just Pax6 concentrations.

Movie 2. 1-D numerical simulations of Model B. Simulations demonstrate that the *Pax6/ Fst/Tgfb2* network of Model B could spontaneously polarise the optic vesicle into Pax6+ and Pax6- 'poles' if Fst:Tgfb2 hetero-tetramer diffusion exceeds that of Fst monomers. Pax6 was initially distributed throughout a circular tissue, but with small local fluctuations. Simulations were performed using different relative diffusion rates for Fst monomers (x-axis) and Tgfb2 dimers (y-axis), while Fst:Tgfb2 hetero-tetramer diffusion rate was held constant. Simulations are shown three times; first with both Pax6 (red) and activated Tgfb receptor (i.e. Tgfb2:Tgfbr complex; cyan) concentrations together; second with just activated receptor concentrations; third with just Pax6 concentrations.
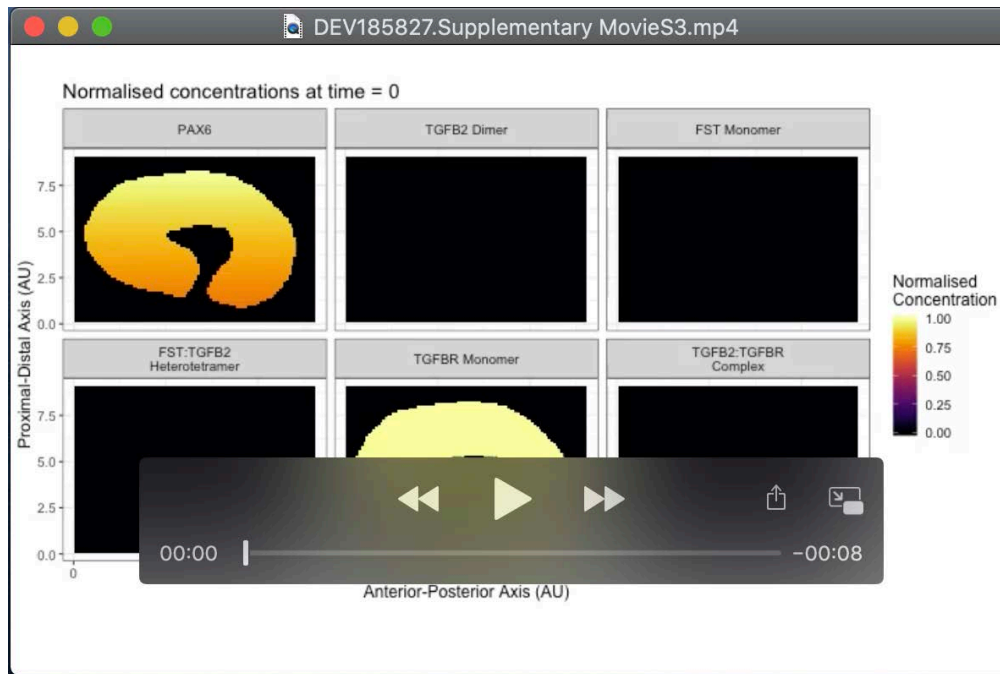
Movie 3. 2-D numerical simulation of Model C. This movie shows a 2-D numerical simulation of Model C within an explant-shaped domain. The initial distal-high to proximal-low Pax6 pattern is dynamically re-polarised, generating a graded expression along the explant's longest axis. This long axis could represent the optic vesicle's former Anterior-Posterior axis (as labelled) or Dorsal-Ventral axis. The movie shows the evolving concentration patterns of all six molecular species represented in the model.

Table S1. Primer sequences for qRT-PCR.

Target Gene	Upstream (5' – 3')	Downstream (5' – 3')
β-actin	CCAGCTGGGAGGAGCCGGT	CTGGGGAACACAGCCCGCTT
Pax6	AGTGGAGAAGGGAGGAGAAG	TGATGTGAAAGAGGAAACGGG
Vsx2	CACTACCCTGATGTGTATGCTAG	CCCATACTCAGCCATCACG
Wnt2b	TGAGTGCCAGTACCAATTCC	GAGATGGCGTAGACGAAGG
Mitf	AAGAACTGGGCACCTTGATAC	TGTCTGTTCTCAAGTTCCTTCG
Tgfb2	TTGATGTCACTGAGGCTGTAC	CAAATCTTGCTTCAGGCTCC
Fst	TGAAAGCAAGTCAGATGAGGC	CCTCTGGGTCTTCGTTAATGG

Table S2. Primer sequences for PCR cloning of in situ hybridization probes.

Target Gene	Upstream (5' – 3')	Downstream (5' – 3')
Vsx2	CTTCCCTCAACCAGAGCAAG	CATGGCATCTGCAAGCTAAA
Mitf	GGGATCCAAACTGGAAGACATC	CCTTTCAAGTGTTTACAAAGAACG
Fst	CAGGCTGGGAACTGCTGG	ACAGGCTGCCTCTTTCAT
Pax6 C-term	AGGAGGGGAGAACACCAACT	GGGAAATGAGACCTGTGGAA

Table S3. Primer sequences for PCR cloning of expression constructs.

Gene	Upstream (5' – 3')	Downstream (5' – 3')
dnPax6	GCCATCGATATGCAGAACAG	TTACTTTTCGCTAGCCAGGTTG
Fst-HA	ATGTTAAATCAGAGGATCCACCCG	CCACTCTAGAATGGAAGATATAGGAAAGC