Hedgehog signaling regulates cell motility and optic fissure and stalk formation during vertebrate eye morphogenesis

Hannah B. Gordon*, Sarah Lusk*, Keith R. Carney, Emily O. Wirick, Brooke Froelich Murray, and Kristen M. Kwan

Department of Human Genetics
University of Utah, Salt Lake City, UT 84112

* denotes equal contribution

Corresponding author:

Kristen M. Kwan
Department of Human Genetics
EIHG 5100
University of Utah
15 North 2030 East
Salt Lake City, UT 84112
phone: 801-585-7541
fax: 801-581-7796
email: kristen.kwan@genetics.utah.edu

Abstract

Establishment of precise 3-dimensional tissue structure is vital for organ function. In the visual system, optic fissure and stalk morphogenesis is a critical, yet poorly understood process, disruptions of which can lead to coloboma, a birth defect causing visual impairment. Here, we use 4-dimensional imaging, cell tracking, and molecular genetics in zebrafish to define cell movements underlying normal optic fissure and stalk formation. We determine how these events are disrupted in a coloboma model in which the Hedgehog receptor *ptch2* is lost, resulting in overactive Hh signaling. In the *ptch2* mutant, cells exhibit defective motile behaviors and morphology. Cells that should contribute to the fissure do not arrive at their correct position, and instead contribute to an ectopically large optic stalk. Our results suggest that overactive Hh signaling, through overexpression of downstream transcriptional targets, impairs cell motility underlying optic fissure and stalk formation, via non-cell autonomous and cell-autonomous mechanisms. More broadly, our cell motility and morphology analyses provide a new framework to study other coloboma-causing mutations disrupting optic fissure or stalk formation.

Introduction

During eye development, cells and tissues must undergo dynamic movements to generate the complex, precise 3-dimensional architecture that is critical for its function. The embryonic structure of the eye is initially established in the process of optic cup morphogenesis: the optic vesicle evaginates as an outpocketing of the neuroepithelium, then elongates and invaginates to generate the optic cup, as the connection between the optic cup and prospective brain narrows to become the optic stalk (Chow and Lang, 2001; Yang, 2004; Adler and Canto-Soler, 2007; Martinez-Morales and Wittbrodt, 2009; Fuhrmann, 2010; Sinn and Wittbrodt, 2013; Bazin-Lopez et al., 2015). This process occurs rapidly in zebrafish, from 10-24 hours post fertilization (hpf). During optic cup formation, a cleft structure, known as the optic fissure, appears at the ventral side of the optic cup and optic stalk (Figure 1A). Subsequently, the optic fissure closes along its entire length in the optic cup and stalk: the margins of the fissure fuse to generate the seamless conduit through which retinal vasculature enters the eye and retinal ganglion cell axons exit. Significant work has focused on later stages of optic fissure development, with an emphasis on optic fissure fusion (24-48 hpf in zebrafish), including genetic and descriptive studies in multiple organisms (Hero, 1989; Hero, 1990; Hero et al., 1991; James et al., 2016; Bernstein et al., 2018). Yet the early stages of optic cup morphogenesis that generate the optic fissure and optic stalk remain a mystery: little is known of the cellular dynamics driving their initial formation.

The molecular mechanisms regulating optic fissure and stalk formation are also poorly understood, yet disruptions to the development of these crucial structures manifest as the birth defect uveal coloboma, a significant cause of pediatric blindness (Gregory-Evans et al., 2004; Fitzpatrick and van Heyningen, 2005). The genetic architecture underlying coloboma is heterogeneous; human genetic studies have uncovered numerous putative causative mutations (Gregory-Evans et al., 2004;

Williamson and FitzPatrick, 2014; Patel and Sowden, 2017). Work in model organisms, (mouse, chick, zebrafish), has begun to examine these genes during eye development, also identifying crucial molecules, signaling pathways, and extraocular cell populations. For example, human mutations in *PAX6*, *PITX2*, and *SOX11* can all result in coloboma, and animal models have uncovered transcriptional network interactions (Gage et al., 1999; Ozeki et al., 1999; Stull and Wikler, 2000; Baulmann et al., 2002; Singh et al., 2002; Azuma et al., 2003; Gregory-Evans et al., 2004; Pillai-Kastoori et al., 2014). Signaling molecules such as Gdf6, Lrp6, and retinoic acid have also been implicated through a combination of human and model organism genetics (Asai-Coakwell et al., 2007; Zhou et al., 2008; Lupo et al., 2011; French et al., 2013). Yet even as genetic models and a growing coloboma gene network continue to emerge, an understanding of how these mutations disrupt the actual underlying morphogenetic events remains elusive.

One pathway vital to optic fissure development is the Hedgehog (Hh) signaling pathway: mutations upstream, within, and downstream of Hh signaling can induce coloboma in humans and model organisms (Gregory-Evans et al., 2004). For example, upstream of Hedgehog signaling, mutations in Sox genes disrupt optic fissure development in zebrafish by altering Hedgehog ligand expression (Pillai-Kastoori et al., 2014; Wen et al., 2015). Additionally, SHH itself can be mutated in human coloboma (Schimmenti et al., 2003). The downstream transcriptional target *PAX2* is mutated in human renal-coloboma syndrome and has been studied in mouse and zebrafish (Keller et al., 1994; Sanyanusin et al., 1995; Favor et al., 1996; Torres et al., 1996; Macdonald et al., 1997; Eccles and Schimmenti, 1999).

The Hh receptor *ptch2* is also associated with coloboma. Human mutations in *PTCH* result in Gorlin Syndrome (Hahn et al., 1996; Smyth et al., 1999); affected individuals can present with coloboma (Ragge et al., 2005). Ptch2 is a negative feedback regulator:

its expression is induced as a downstream transcriptional target of Hedgehog signal transduction, and the protein inhibits signaling by the transmembrane molecule Smoothened. Therefore, loss-of-function mutations in *ptch2* result in overactive Hedgehog signaling specifically within cells responding to Hedgehog ligand. In zebrafish, the loss-of-function *ptch2*^{tc294z} mutant (Lee et al., 2008) exhibits coloboma (Figure 1B,C). Rescue experiments using the Hedgehog signaling inhibitor cyclopamine demonstrated that coloboma is caused by overactive Hedgehog signaling (Lee et al., 2008), however, the cellular and molecular mechanisms by which this disrupts optic fissure development remain unknown.

Optic fissure morphogenesis, a multi-stage process including formation and fusion, could potentially be disrupted at any step to result in coloboma. Additionally, the optic stalk, through which the optic fissure extends, is itself a poorly understood structure that is critical for the visual system. Here, we set out to directly visualize and determine the cellular events underlying the initial step of optic fissure and stalk formation. What cell movements are involved? How is this disrupted in a specific coloboma model of overactive Hh signaling? Defining the basic cellular processes provides a framework to begin to understand how these structures form and develop. Further, this will lay the groundwork for dissecting additional coloboma-causing mutations and establishing the spectrum of cellular events that are sensitive to genetic perturbations.

Here, using a combination of 4-dimensional microscopy, computational methods, and molecular genetics, we define the cell movements underlying normal optic fissure and stalk formation; determine the morphogenetic defects in the *ptch2*^{tc294z} mutant, in which optic fissure and stalk formation are disrupted; and examine the molecular basis by which overactive Hedgehog signaling causes these defects.

Methods

Zebrafish husbandry and mutant/transgenic lines

All zebrafish husbandry (*Danio rerio*) was performed under standard conditions in accordance with University of Utah Institutional Animal Care and Use Committee (IACUC) Protocol approval (Protocol #18-02006). Embryos (Tü or TL strains) were raised at 28.5-30°C and staged according to time post fertilization and morphology (Kimmel et al., 1995). Mutant lines were previously described: *ptch2*/*blowout*^{tc294z} (Karlstrom et al., 1996; Koudijs et al., 2008; Lee et al., 2008); *gli1/detour*^{ts269} (Brand et al., 1996; Karlstrom et al., 1996).

For genotyping: genomic DNA was extracted from single embryos or adult fins, incubated at 95°C in 0.05 M NaOH for 30 minutes, then neutralized with 1M Tris pH 8.0. The *ptch2* locus was genotyped via one of two methods. A CAPS assay (Konieczny and Ausubel, 1993) with the following primers was used: ptch2_F: 5'-CCATGATAAGTACGACACCACTGGAGAG-3', ptch2_R: 5'-CACTACACCAAATCCCTGATGGATGG-3'. The *ptch2*^{tc294z} mutation creates an AvaII cut site; the wt allele is uncut. Alternatively, we developed an HRMA protocol (Parant et al., 2009) with the following primers: ptch2HRMA_F: 5'- CTGCACCTTCCTGGTGTGTG-3', ptch2HRMA_R: 5'- GGTAGAAATGGATTAGAGTGAGAGGAA-3'. These primers generate a 96 bp amplicon; homozygous wt and mutant duplexes melt at 87.6ºC and 88.5ºC, respectively. The *gli1* locus was genotyped using a dCAPS assay (Neff et al., 1998) with the following primers: gli1_F: 5'-GTGCCAGCGATCCGGTGCGATCC-3', gli1_R: 5'- CATTCCTGCACCCTGGTATTGCATCC-3'. The mutant allele combined with the dCAPS primer creates a cut site for HpyCH4III and the wt allele is uncut.

RNA synthesis and nucleic acid injections

Capped RNA was synthesized using pCS2 templates (pCS2-EGFP-CAAX, pCS2FA-H2A.F/Z-mCherry, pCS2FA-nls-Kaede, pCS2-rSmoM2-EGFP, pCS2FA-Kaede, pCS2-EGFP-Xcentrin, pCS2FA-mCherry-CAAX, pCS2FA-Tol2tranposase) and mMessage mMachine SP6 kit (Ambion). RNA was purified (Qiagen RNeasy Mini Kit) and ethanol precipitated. For fluorescent proteins, 300-500 pg RNA was injected into the cell of 1-cell embryos. For activated Smoothened, 175-200 pg rSmoM2-EGFP RNA was injected into the cell of 1-cell embryos. To label centrosomes, 200 pg EGFP-Xcentrin RNA and 200 pg mCherry-CAAX RNA were injected into the cell of 1-cell embryos. For mosaic labeling of cells (for cell morphology quantification), 25 pg of pCS2-mCherryCAAX DNA was injected into the cell of 1-cell embryos.

Transgenic constructs and establishment of stable lines

Transgenic line with ubiquitous expression of membrane-bound GFP, Tg(*bactin2*:EGFPCAAX)^{z200}: the bactin2 promoter (p5E-bactin2) was recombined with a middle entry clone of EGFP-CAAX (pME-EGFPCAAX) and a 3' clone of the SV40 late polyA signal sequence (p3E-polyA) into the Tol2 transposon-flanked destination vector, pDestTol2pA2 (Kwan et al., 2007).

Transgenic line marking Shh-producing cells, Tg(*SHH*:H2A-EGFP)^{z205}: the -2.2 kb *Shh* promoter and the downstream enhancer elements (arABC) were a kind gift from Ferenc Müller, Yavor Hadziev, and Uwe Strähle (Chang et al., 1997; Muller et al., 1999; Ertzer et al., 2007). The -2.2 kb *Shh* promoter was cloned into a multisite Gateway 5' entry vector, and the arABC elements were cloned into a multisite Gateway 3' entry vector. These two pieces were used with a Gateway middle entry clone of H2A-EGFP-

polyA (pME-H2A-EGFP-pA) to generate the full transgene expression construct in the Tol2 transposon-flanked destination vector, pDestTol2pA2 (Kwan et al., 2007).

Transgenic line marking Hh-responding cells, Tg(GBS-*ptch2*:H2A-mCherry)^{z206}: the 900 bp *ptch2* promoter containing an additional Gli recognition sequence (in a 5' Gateway entry vector) was a kind gift from Rolf Karlstrom (Shen et al., 2013). This was recombined with a middle entry clone of H2A-mCherry and a 3' entry clone of the SV40 late polyA signal sequence to generate the full transgene expression construct in the Tol2 transposon flanked destination vector, pDestTol2pA2 (Kwan et al., 2007).

Assembled transgene expression constructs were coinjected with *Tol2* transposase (30 pg DNA + 25 pg transposase RNA) into the cell of a 1-cell embryo. Injected embryos were screened for fluorescence reporter expression at 24 hpf and raised to adulthood.

Coloboma scoring

Embryos were individually screened and scored for coloboma at 52-55 hpf using an Olympus SZX16 stereomicroscope. Eyes that exhibited abnormal morphology outside of the coloboma phenotype were excluded from final analysis. Viewing the back of the eye being scored, and focusing at the depth of the RPE, embryos that scored as positive for coloboma displayed an expanded region lacking pigmentation in the area of the optic nerve head; in a positive embryo, this area was distinctly wider and more open than the rest of the optic fissure that was undergoing fusion at the ventral side of the optic cup. Pigmentation and fissure fusion along the ventral surface of the optic cup can show variability with timing among both wild type and *ptch2*^{tc294z} mutant embryos, therefore, simple (often subtle) expansion of a hypopigmented region along the ventral surface of the optic cup (away from the optic nerve head) was not sufficient. Embryos that were not clearly positive for coloboma were counted as negative.

All genetic experiments and pharmacological treatments were blindly scored. Embryos were subsequently genotyped as described above.

Antibody staining

Embryos were raised until the stage of interest and fixed in 4% paraformaldehyde (Electron Microscopy Science, 15710) overnight at 4°C. Embryos were permeabilized in PBST (PBS with 0.1% Triton X-100) and blocked in 2% BSA in PBST at room temperature. For primary antibody: anti-Pax2a (1:200; Genetex, GTX128127), anti-aPKC (1:100; Santa Cruz Biotechnology, sc-216), anti-phospho-histone H3 (1:500; Abcam, ab14955) was diluted in blocking solution and incubated overnight at 4°C. For secondary antibody: Alexa Fluor 488 goat anti-rabbit (Life Technologies, A-11008), Alexa Fluor 568 goat anti-rabbit (Life Technologies, A-11011), or Alexa Fluor 568 goat anti-mouse (Life Technologies, A-11001) was co-incubated with 1 µM TO-PRO-3 iodide (Life Technologies, T3605). Embryos were cleared in 70% glycerol for imaging.

Imaging

For timelapse imaging, embryos were dechorionated at the relevant developmental stage, embedded in 1.6% low melting point agarose (in E2+gentamycin) in Delta T dishes (Bioptechs (#0420041500C)). Images were acquired using a Zeiss LSM710 laser scanning confocal microscope. E2+gentamycin was overlaid, and the dish covered to prevent evaporation. No stage heater was used. Based on when the timing of lens separated from the overlying ectoderm, we estimate that sample temperature was very close to 28.5°C. All imaging was performed with a 40X water-immersion objective (1.2 NA).

For cell tracking, 4-dimensional datasets were acquired with the following parameters: 60-63 z-sections, 2.1 µm z-step, 2.75 minutes between z-stacks. For imaging cell transplantation experiments, 4-dimensional datasets were acquired with the following parameters: 60-100 z-sections, 2.1 µm z-step, 6-12 minutes between z-stacks.

Kaede photoconversion was performed using the Zeiss Zen software to select and expose a region of interest (ROI) to 405 nm light for 15-20 seconds. Photoconversion efficiency was assayed by loss of green and gain of red fluorescence in the ROI.

Image analysis: 4D cell tracking/visualization

Cell tracking was performed according to (Kwan et al., 2012), using LongTracker. Briefly, 4-dimensional data sets were pre-processed as previously described (Kwan et al., 2012), linearly adjusting brightness and contrast. LongTracker was used to select nuclei at specific time points and *z* positions, while simultaneously viewing *xz* and *yz* reslices and adjacent frames in *z* and *t*, stepping through the data set forward or backward in time. For accurate cell tracking, we required at least 50% nuclear pixel overlap between time points. Data were exported as a spreadsheet of nuclear centroid position, nuclear TIFF images, or 3-dimensional trajectories. Trajectories were checked for discontinuities in 4 dimensions using FluoRender (Wan et al., 2012). Therefore, tracking was performed and checked in 4 dimensions. Figures show a dorsal view of the tracking data, and the pseudocolored signal is the actual nuclei that were tracked. Trajectories are generated by summing the nuclear masks over time.

Image analysis: optic fissure opening angle

3-dimensional data sets of live embryos labeled for membranes (EGFP-CAAX) were oriented in FluoRender (Wan et al., 2012) for a lateral view. In this lateral view, we focused in on the midpoint of the lens, using the lateral cutaway tool. TIFF images of this view were captured from FluoRender, and the opening of the optic fissure was measured using the angle tool in Fiji. The rays of the angle projected along the optic fissure margins, with the vertex positioned at the center of the lens.

Image analysis: region/area quantification (Shh reporter, pax2)

Reporter extent measurement: embryo orientation was adjusted using FluoRender to ensure and standardize a dorsal view. Typically, a “trail” of reporter expression was observed along the lateral edge of the optic vesicle. The medial and lateral anterior-posterior extent of reporter expression were measured; the anterior-posterior midpoint was then calculated and graphed as a ratio of the entire optic vesicle length.

Reporter fluorescence intensity measurement: measurements were performed on a single slice at the dorsal-ventral midpoint of the optic vesicle. Images were rotated in ImageJ/Fiji, such that the optic vesicle was oriented vertically. A rectangular ROI was drawn (using the rectangle tool): the width of the rectangle was 44 pixels (30.44 µm), and the length of the rectangle spanned the anterior-posterior length of the optic vesicle. Using a vertical profile, the command “Plot Profile” was used to quantify fluorescence intensity at each vertical pixel position along the entire rectangle. Average fluorescence intensity measurements were calculated for the anterior and posterior quarters of the rectangle; for each embryo, the relative fluorescence intensity measurement was normalized as the ratio of the anterior quarter to the posterior quarter.

Pax2a extent measurement: embryo orientation was adjusted using FluoRender to ensure and standardize a dorsal view. The anterior-posterior extent of Pax2a staining was measured as a ratio of entire optic vesicle length.

Image analysis: cell morphology and orientation angle quantification (Kaede, mosaics and transplants)

Roundness measurement: embryos were oriented dorsally in FluoRender, and TIFF images captured of dispersed cells in the stalk, for which morphology was easily visualized. An outline was drawn around the cell using the freehand tool in Fiji/ImageJ, and roundness was measured (as one of the shape descriptors in the Measurements).

Orientation angle: similar to Roundness, embryos were oriented dorsally in FluoRender, and TIFF images captured of dispersed cells in the stalk, for which morphology was easily visualized. The angle tool in Fiji/ImageJ was used: one ray extended along the anterior-posterior axis of the embryo, and the other ray extended along the long axis of the cell.

Image analysis: 3-dimensional cell counting (custom MATLAB script, "Abacus")

Nuclear signal in 3-dimensional data sets was used for cell counting.

Preprocessing: Nuclear signal was in 8-bit gray level resolution. A Hi-Lo lookup table (ImageJ/Fiji) was used to linearly adjust the signal such that the background noise level was set to zero, and the brightest pixels were just below 255.

Abacus algorithm (also see Supplementary File 1 for MATLAB code): the image z-stack with pixel size = dx and z-step = dz) was low-pass filtered using a 3D Gaussian smoothing kernel with a standard deviation in the z direction (σ_z) that is proportional to the standard deviation in the xy-plane (σ_{xy}) such that $\sigma_z = \sigma_{xy}(dx/dz)$. A threshold value, L, was set to eliminate background noise, and several linearly spaced isosurfaces were calculated to visually represent the outer edge of the fluorescent nuclear markers. Using the filtered z-stack, regional maxima with a minimum height h, were located and the centroids of those regions calculated. Centroid markers were overlaid with the isosurfaces to visually verify if the centroid locations correlate with the centers of the nuclear marker outlines. The parameters σ_{xy}, σ_z, L, and h were adjusted until the centroid markers matched the center locations of the nuclear marker isosurfaces. Visual verification of the centroid and isosurface overlay was performed by stepping through narrow cross-sectional views of the data.

Box and whisker plots

Box and whisker plots were generated using the ggplot2 package in R. The band inside the box is the median. The upper and lower "hinges" correspond to the first and third quartiles. The upper whisker extends from the upper hinge to the highest value within (1.5 * IQR), where IQR is the inter-quartile range. The lower whisker extends from the lower hinge to the lowest value within (1.5 * IQR). Data points outside of the ends of the whiskers are outliers.

Blastula stage transplantation and analysis

Donors or hosts for the wild type sibling condition were either embryos from incrosses of Tü wild types, or heterozygote siblings from crosses of *ptch2*^{tc294z} heterozygotes and homozygotes (identified by genotyping after the experiment). Donor or host embryos for the mutant condition were from crosses of *ptch2*^{tc294z} heterozygotes and homozygotes (identified by genotyping after the experiment). Host embryos were additionally Tg(*bactin2*:EGFPCAAX)^{z200}; they were marked with ubiquitous expression of EGFP-CAAX.

Donor embryos were injected at the 1-cell stage with 1 nl Alexa-568 dextran (Thermo Scientific #D22912). At the mid-blastula stage, donor and host embryos were dechorionated and transferred to agar-coated dishes in E3 + 1% pen/strep (Invitrogen, #15070063). Up to 50 cells were removed from donor embryos, and approximately 10-15 cells transferred to each of four host embryos. Donor embryos were immediately subjected to genomic DNA extraction and genotyping, while host embryos were transferred to agarose-coated wells of a 24-well culture dish. At ~11 hpf, host embryos were screened for donor cells in the eye region and mounted for timelapse imaging. Embryos were screened for a second time under the confocal microscope for donor cells residing within the midline region, the area from which prospective nasal optic fissure cells arise. After timelapse imaging, embryos were de-embedded and subjected to genotyping.

Transplantation analysis was carried out specifically on cells that initially resided in the region of interest: the midline region, from which the nasal optic fissure and optic stalk are derived. Cells were scored for final position (optic fissure/cup or optic stalk), roundness, and orientation angle.

Pharmacological experiments

Fertilized embryos were grown until 5.5 hpf, at which time they were dechorionated and placed into agarose-coated wells of a 6-well culture dish. Embryos were cultured in 7 ml of E3 with indicated concentrations of BMS-833923 (Cayman Chemical #16240), PP2 (Calbiochem #529573), SU6656 (Calbiochem #572635), or DMSO (Sigma #D8418), based on published reports in zebrafish (Molina et al., 2007; Murphy et al., 2011; Gallardo et al., 2015; Armstrong et al., 2017), as well as our own dose response experiments. At 13 hpf (8 somite stage), embryos were removed from pharmacological treatment, washed once, and transferred to fresh agarose-coated dishes containing E3. Coloboma scoring was performed blind at 55 hpf, and embryos subsequently genotyped.

Results

Optic cup morphogenesis and optic fissure and stalk formation are disrupted in the ptch2^{tc294z} mutant

The cellular events underlying normal optic fissure and stalk formation have been unknown. To determine the cellular basis of their formation, we acquired 4-dimensional timelapse imaging data sets of optic cup morphogenesis in wild type embryos, spanning 12-24 hours post fertilization (hpf). Embryos were labeled for membranes (EGFP-CAAX) and chromatin (H2A.F/Z-mCherry). In wild type embryos, tissue rearrangements are executed (Figure 1D-G; Movie 1), many of which have been described by multiple groups (Martinez-Morales and Wittbrodt, 2009; Picker et al., 2009; Kwan et al., 2012; Heermann et al., 2015; Sidhaye and Norden, 2017). These tissue rearrangements lead

to the formation of the optic cup at 24 hpf, comprised of neural retina, retinal pigmented epithelium (RPE), and lens. At this timepoint, we can visualize 3-dimensional optic cup structure by manually segmenting the neural retina and RPE (*teal*), lens (*gray*), and optic stalk (*gold*) (Figure 1H). The optic fissure is apparent as a cleft structure with two closely apposed margins (temporal and nasal) at the ventral side of the optic cup (Figure 1H, *arrowhead*; Movie 2).

The loss-of-function *ptch2*^{tc294z} mutant displays coloboma, a structural defect in the eye due to aberrant development of the optic fissure (Lee et al., 2008). Although the morphological features of the phenotype were described in detail at stages after optic cup morphogenesis, the precise timing of underlying cellular defects have been unknown. We first asked whether optic fissure and stalk formation might be disrupted in the *ptch2*^{tc294z} mutant, by performing 4-dimensional imaging of optic cup morphogenesis. In the *ptch2*^{tc294z} mutant, we observe no gross defect in optic vesicle formation; the volumes of mutant and wild type optic vesicles are not significantly different (Figure 1N; wt 0.79±0.1x10^6 μm^3, mutant 0.71±0.1x10^6 μm^3). As optic cup morphogenesis progresses, the eye appears (in single optical sections) to form normally (Figure 1I-L; Movie 3), with neural retina and RPE enwrapping the lens. However, 3-dimensional visualization of the optic cup reveals already apparent defects in the ventral portion of the eye and optic stalk. Whereas the wild type optic cup contains closely apposed optic fissure margins (Figure 1H, *arrowhead*), the *ptch2*^{tc294z} mutant optic fissure does not (Figure 1M, *arrowhead*; Movie 4). Quantification reveals that the optic fissure opening is significantly larger in *ptch2*^{tc294z} mutant embryos (Figure 1Q; wt 19±5.25°, mutant 59.6±6.2°). In addition to this structural defect, the mutant optic cup is smaller than wild type (Figure 1O; wt 1.89±0.13x10^6 μm^3, mutant 1.27±0.11x10^6 μm^3), and the mutant optic stalk is significantly larger (Figure 1O; wt 0.18±0.05x10^6 μm^3, mutant

0.51±0.08x10^6 µm^3). The total volume of the mutant optic cup and stalk is significantly smaller than wild type (Figure 1O; wt 2.07±0.17x10^6 µm^3, mutant 1.77±0.1x10^6 µm^3), yet despite the differences in total volume, the optic stalk is still a larger relative volume (Figure 1P; wt 0.085±0.016, mutant 0.287±0.045). The appearance of this volume defect suggests that some aspect of optic cup morphogenesis is disrupted in the *ptch2*^{tc294z} mutant.

In other contexts, Hedgehog signaling can drive cell proliferation (Evangelista et al., 2006), and hyperproliferation can be sufficient to cause coloboma (Kim et al., 2007). However, *ptch2* mutants were not previously found to exhibit increased proliferation, at stages subsequent to optic cup formation (Lee et al., 2008). Because we observed morphological defects prior to when proliferation had been previously assayed, we tested whether mitosis might be altered earlier (24 hpf). We find no significant difference in the number of phospho-histone-H3-positive cells in the optic cup or stalk in *ptch2*^{tc294z} mutants (Figure 1R,S; wt optic cup 85±15, mutant optic cup 82±3; wt stalk 16±4, mutant stalk 20±4). Therefore, changes in proliferation are less likely to account for the difference in eye volume observed in the *ptch2*^{tc294z} mutants; defects may be due to altered morphogenetic movements.

Finally, the coloboma phenotype in *ptch2*^{tc294z} mutants is incompletely penetrant at 2 days post fertilization (Lee et al., 2008), so we assayed penetrance of the 24 hpf phenotype, in particular, the large optic stalk and its relationship to the appearance of coloboma. Embryos were imaged at 24 hpf, then scored blindly for coloboma at 55 hpf. We find that the optic stalk phenotype at 24 hpf is fully penetrant, while coloboma is incompletely penetrant; all embryos with coloboma at 55 hpf previously exhibited a large optic stalk (Figure 1T). We conclude from these data that optic cup morphogenesis, optic stalk formation, and optic fissure formation are defective in *ptch2*^{tc294z} mutants, and

that optic fissure and stalk defects are apparent prior to optic fissure fusion. Thus, we focused on how these phenotypes originate, at this early stage of development.

Cell movements underlying optic fissure and stalk formation are disrupted in the ptch2^{tc294z} mutant

Because the *ptch2*^{tc294z} mutant displayed defective optic fissure and stalk formation, we sought to identify the specific cell movements disrupted during this process, using two independent assays.

First, we utilized a previously-developed photoactivation-based fate mapping assay using nls-Kaede, a nuclear-localized version of the photoconvertible fluorophore Kaede (Ando et al., 2002; Kwan et al., 2012). This assay allows us to visualize the last cells to move into the optic vesicle during evagination, during a period we termed extended evagination (Kwan et al., 2012). These cells contribute directly to the nasal margin of the optic fissure in wild type embryos (Kwan et al., 2012). At 12 hpf, we photoconverted the entire optic vesicle (and some overlying ectoderm) to red fluorescence (pseudocolored magenta) (Figure 2A). At 24 hpf, we visualize extended evagination as green nuclei within the optic cup, as these cells have moved out of the midline region and into the eye after 12 hpf. In wild type embryos, green nuclei are found in the nasal margin of the optic fissure, the ventronasal retina, and optic stalk (Figure 2B,C). Green nuclei in the lens are derived from ectodermal cells not photoconverted. In the *ptch2*^{tc294z} mutant, these late-moving cells appear excluded from the defective optic fissure (Figure 2D-F). Green nuclei are still found in the nasal retina (dorsal to the optic fissure) and optic stalk, indicating that some cell movement occurred, but not into the optic fissure. The temporal margin still remains devoid of green nuclei. We quantified the proportion of green cells within the optic cup under wild type and *ptch2*^{tc294z} mutant conditions, using a custom MATLAB-based 3-dimensional cell counting script Abacus

(see Methods, Supplementary File 1). The *ptch2*^{tc294z} mutant optic cup contains a significantly smaller proportion of green nuclei in the optic cup (Figure 2J; wt 18.9±0.5%, mutant 8.7±1.7%), indicating impaired extended evagination cell movements. As an independent method to overactivate Hedgehog signaling, we injected RNA encoding an activated form of the Smoothened (Smo) transmembrane protein (rSmoM2-EGFP, (Huang and Schier, 2009)). Expression of activated Smo also impairs extended evagination cell movements (Figure 2G-I): no green nuclei are found in the optic cup or optic stalk, a more severe phenotype than the *ptch2*^{tc294z} mutant. Consistent with this, the optic fissure opening is also more severely affected by activated Smo (wt 19±5.3°; *ptch2*^{tc294z} mutant 59.6±6.2°; activated Smo 88.4±13.4°; data not shown, Figure 1Q). This could reflect functional differences in the manipulations: *ptch2* is a negative feedback regulator whose loss leads to overactive Hedgehog signaling, but only within cells responding to Hedgehog ligand. In contrast, activated Smo RNA injection induces overactive Hedgehog signaling in all cells, even those not responding to Hedgehog ligand. Yet both independent methods of overactivating Hh signaling disrupt cell movements contributing to the optic fissure.

These fate mapping experiments provide initial indications of defects in the *ptch2*^{tc294z} mutant. In order to identify precisely when and where morphogenetic movements are disrupted, we utilized a second assay to examine cell movements: 4-dimensional cell tracking using our MATLAB-based manual cell tracking software, LongTracker (Kwan et al., 2012). Since the cellular basis of normal optic fissure and stalk formation has been unknown, we first mapped the origins and trajectories of cells contributing to the optic fissure margins in the optic cup and the optic stalk. LongTracker allows us to carry out retrospective cell tracking; therefore, we selected cells in the optic fissure margins (nasal and temporal) and optic stalk at optic cup stage (Figure 3B). We tracked the movement

of these nuclei backwards in 4 dimensions to determine their origins at optic vesicle stage (12 hpf). In wild type embryos, cells contributing to the optic fissure margins and stalk arise from adjacent domains at optic vesicle stage (Figure 3A). There is a key spatial difference: the nasal margin (*red/yellow shades*) and optic stalk cells (*purple shades*) are found within the midline domain, while the temporal margin cells (*blue shades*) are found within the optic vesicle. This is consistent with our nls-Kaede results: nuclei contributing to the temporal margin of the optic fissure are converted to red/magenta fluorescence at 12 hpf, while nuclei contributing to the nasal margin and optic stalk, arising from the midline region, remain green. Having mapped the origins of these cells, we visualized trajectories of prospective optic fissure and optic stalk cells. All three groups of cells move along a "J-shaped" trajectory, with nasal margin cells moving into the optic vesicle and following temporal margin cells. Optic stalk cells subsequently follow nasal margin cells (Figure 3D-G,D'-G'; Movie 5,6).

With these wild type data in hand, we asked when and where optic fissure and stalk cell movements are affected in the *ptch2*^{tc294z} mutant. Using the wild type data, we first generated a fate map of our regions of interest (Figure 3C; temporal margin, nasal margin, and optic stalk), and applied it to the *ptch2*^{tc294z} mutant data set (Figure 3H,I), reasoning that forward cell tracking from those domains would allow us to pinpoint defective cell movements. We find that cells arising from wild type optic stalk, nasal, and temporal domains at optic vesicle stage do not achieve their correct positions in the *ptch2*^{tc294z} mutant optic cup (Figure 3I,J). Cells that originate from the prospective temporal domain instead contribute to the nasal side of the disrupted *ptch2*^{tc294z} mutant optic fissure; cells from nasal and stalk origins are intermingled within the stalk; these positions are shifted proximally toward the midline relative to wild type. To determine whether the final positions of these cells were incorrect due to aberrant trajectories, we visualized their movements: mutant cells travel along a shallow J-shaped trajectory, with

cells from the prospective temporal region (*blue shades*) followed by cells from the prospective nasal and stalk regions (Figure 3K-N,K'-N'; Movie 7,8; *red/yellow shades* and *purple shades*). Therefore, trajectory shape is not grossly changed.

Although the mutant trajectory shape appears superficially normal, we quantified individual cell movements (all in 3-dimensions) for average speed, trajectory length, and net displacement. In the *ptch2*^{tc294z} mutant, cells from each domain move significantly faster than their wild type counterparts (Figure 3O; wt temporal 0.56±0.02 µm/min, mutant temporal 0.62±0.01 µm/min; wt nasal 0.55±0.03 µm/min, mutant nasal 0.71±0.1 µm/min; wt stalk 0.46±0.03 µm/min, mutant stalk 0.69±0.09 µm/min). Mutant cells also execute a significantly longer total 3-dimensional trajectory length (Figure 3P; wt temporal 432.8±17.6 µm, mutant temporal 457.3±6.7 µm; wt nasal 422.9±20.5 µm, mutant nasal 524.9±75.8 µm; wt stalk 355.5±23.1 µm, mutant stalk 507.2±64.4 µm). However, 3-dimensional net displacement is not significantly different in any domain (Figure 3Q; wt temporal 132.4±10.3 µm, mutant temporal 120.8±20.8 µm; wt nasal 137.1±5.7 µm, mutant nasal 127.4±13.8 µm; wt stalk 121.2±18.3 µm, mutant stalk 124±26.5 µm). We hypothesize that processive cell motility may be affected in the *ptch2*^{tc294z} mutant optic fissure and stalk: cells move faster and over a longer total path, but do not achieve the same destination as their wild type counterparts.

The tracked cells from the 3 domains all contributed to the nasal side of the optic fissure and stalk, therefore, we asked from where the mutant temporal margin arises. We selected nuclei of temporal margin cells from the *ptch2*^{tc294z} mutant optic fissure at 24 hpf and tracked them in 4-dimensions retrospectively to their origin at 12 hpf (Figure S1A,B). These cells arise from a position which, in wild type, should contribute to temporal retina (Kwan et al., 2012), but they move along a novel trajectory (Figure S1C-F,C'-F'; Movie 9,10). These cells may either be directly affected by overactive Hedgehog signaling, or indirectly affected as a secondary consequence of other

disrupted cell movements; to resolve these possibilities, we assay localization of overactive Hh signaling later in the manuscript.

Taken together, our data indicate that cell movements underlying optic fissure and stalk formation are disrupted in the *ptch2*^{tc294z} mutant. No fissure cells achieve their correct final positions, and notably, nasal fissure cells contribute to the stalk, instead of the optic cup.

Single cell behaviors and morphology are disrupted in the ptch2^{tc294z} *mutant*

The data presented thus far suggest that cell motility is affected during *ptch2*^{tc294z} mutant optic fissure and stalk formation. To directly observe motile behaviors and morphologies, we first used cytoplasmic Kaede. We sought to visualize cell behavior in the prospective nasal margin; in the *ptch2*^{tc294z} mutant, these cells aberrantly contribute to the optic stalk instead of optic cup. Using our cell tracking data and fate map, we photoconverted prospective nasal margin and some optic stalk cells within the midline region at 12 hpf. In wild type embryos, cells exit the midline region, adopt an elongated morphology, move through the forming optic stalk, and into the retina (Figure 4A-F; Movie 11), where they contribute to the ventronasal retina and nasal margin of the optic fissure (Figure 4B,F; *arrowhead*; Figure S2A,A'). A few optic stalk cells are also marked (Figure 4F).

In *ptch2*^{tc294z} mutant embryos, when the same midline region is photoconverted, we observe altered cell behaviors and morphologies (Figure 4G-L; Movie 12). Initially, cells appear normal, moving out of the midline region and appearing elongated. The first cells to exit the midline continue to exhibit wild type behaviors, moving through the optic stalk and into the optic cup (Figure 4L; *arrowhead*). These cells contribute to nasal retina, but dorsal to the optic fissure. However, the majority of cells take on an aberrant, less elongated morphology and exhibit motile behaviors within the optic stalk (Figure 4L;

asterisk; Figure S2B,B'). No marked cells contribute to the disrupted optic fissure, consistent with the nls-Kaede experiment and cell tracking analyses (Figure 2F, 4H).

We quantified the changes in cell morphology within the optic stalk, using mosaic labeling via plasmid DNA injection (pCS2-mCherryCAAX) to visualize individual stalk cells (Figure S2C-D'). First, we quantified roundness (Figure 4M), a measure of an object's aspect ratio; a perfect circle has a roundness of 1.0, while an extremely elongated ellipse would have a roundness approaching zero. In wild type embryos, stalk cells are elongated (0.21±0.06); in *ptch2*^{tc294z} mutants, stalk cells are significantly less so (0.39±0.13). Second, we quantified elongation angle, that is, the orientation of cells with respect to the anterior-posterior axis of the embryo (Figure 4N). In wild type embryos, stalk cells are elongated within a narrow range of angles with respect to the anterior-posterior axis of the embryo; the average cell is elongated closely along the anterior-posterior axis. In *ptch2*^{tc294z} mutant embryos, stalk cells are elongated more variably with respect to the anterior-posterior axis; the variance of elongation angles is significantly greater in the *ptch2*^{tc294z} mutant than wild type (P=1.06x10^{-7}).

The greater variance in *ptch2*^{tc294z} mutant stalk orientation angle led us to ask whether tissue polarity is affected: variable cell orientation could be due to disrupted epithelial polarity. Thus, we performed antibody staining for the apical epithelial marker aPKC. In wild type embryos, aPKC labels the contiguous apical surface of the optic cup, optic stalk, and prospective brain (Figure 4O, Movie 13). In *ptch2*^{tc294z} mutant embryos, aPKC still marks an intact apical domain, though the stalk apical surface appears expanded (Figure 4O,P; *yellow arrows*; Movie 14). This is in contrast to other mutants, such as *lama1*, in which apical polarity markers do not mark a coherent surface (Bryan et al., 2016). We further sought to assay cell polarity in the stalk with single cell resolution. To this end, we marked centrosomes via RNA injection of an EGFP-tagged Xcentrin construct (Sepich et al., 2011). We find that in both wild type and *ptch2*^{tc294z}

mutant embryos, individual centrosomes are oriented toward the apical surface of the optic stalk (Figure S2E-F'). Therefore, disruption of cell polarity does not underlie the increased variability in stalk cell orientation in *ptch2*^{tc294z} mutants.

Our analyses reveal motile behaviors and cell morphologies underlying normal optic fissure and stalk formation; in the *ptch2*^{tc294z} mutant, we have identified defects in motility and morphology directly within these cells.

Loss of ptch2 acts via both cell-autonomous and non-cell-autonomous mechanisms to disrupt optic fissure and stalk cell movements

Having defined the cellular events underlying normal optic fissure formation and precise defects in the *ptch2*^{tc294z} mutant, we asked in which cells the *ptch2*^{tc294z} mutation acts to impair cell movements. Does this mutation act directly within migrating prospective optic fissure cells to alter behavior; or does it act in surrounding cells, within the milieu, to alter motile behaviors of neighboring prospective optic fissure cells?

To distinguish between these possibilities, we utilized blastula cell transplantation (Figure 5A). At blastula stages, cells were transplanted from donor embryos (injected with fluorescent dextran) to transgenic host embryos (Tg(*bactin2*:EGFPCAAX)^{z200}), in which membrane-bound EGFP ubiquitously labels cell membranes. Timelapse microscopy was carried out on transplanted embryos in which donor cells were found in the midline, the prospective nasal optic fissure and optic stalk region at 12 hpf. We quantified donor cell behaviors using three criteria which distinguish wild type and mutant cell phenotypes: 1) final position (optic stalk or nasal optic cup/fissure); 2) roundness; and 3) elongation angle.

Control transplantation experiments were performed (Figure S3; Table 1). When wild type cells are transplanted into a wild type host, donor cells contribute more frequently to optic cup and fissure than to stalk; exhibit wild type cell behaviors and

elongated morphologies; and stalk cells are oriented closely along the anterior-posterior axis of the embryo (Figure S3A-E, Figure 5L-N, Movie 15). When mutant cells are transplanted into a mutant host, donor cells contribute more frequently to optic stalk than optic cup/fissure (wt 25% stalk, mutant 66% stalk); exhibit a less elongated morphology (wt roundness 0.28, mutant 0.48); and are oriented with more variability with respect to the anterior-posterior axis (Figure S3F-J, Figure 5L-N, Movie 16). These measurements are statistically significant between these control transplants (Figure 5L-N).

When *ptch2*^{tc294z} mutant donor cells are transplanted into a wild type host, mutant cells move out of the midline region, through the optic stalk, and into the optic cup (Figure 5B-F; Movie 17). These mutant cells (n=7 transplants, 23 cells scored) contribute to both optic cup/fissure and optic stalk, with a frequency intermediate to, and not significantly different from, either control condition (43.5% stalk, Figure 5L, Table 1). Mutant cells maintain an elongated, wild-type morphology (roundness 0.31, Figure 5F,M; *white arrow*), significantly different from mutant control; and are elongated along a narrow range of angles with respect to the anterior-posterior axis, not significantly different from either control (Figure 5N). We interpret these data to suggest that in this condition, there may be cell-autonomous and non-cell-autonomous factors influencing final donor cell position and orientation angle, but cell elongation is influenced largely by non-cell-autonomous factors.

We next analyzed wild type donor cells transplanted into a *ptch2*^{tc294z} mutant host. These cells (n=7 transplants, 41 cells scored) move out of the midline region and contribute primarily to the optic stalk (Figure 5G-K, *yellow arrow*; Movie 18), with a frequency similar to mutant control and significantly different from wild type control (78% stalk, Figure 5L, Table 1). These cells are less elongated, similar to mutant control and significantly less than wild type control (roundness 0.43, Figure 5M), and they are oriented along a wide range of angles, similar to the mutant control, and significantly

different from the wild type control (*P*=0.04, Figure 5N). Under these conditions, wild type cells appear more similar to mutant cells. This suggests, in this condition, a significant role for non-cell-autonomous mechanisms regulating these behaviors.

Taken together, our data suggest that a *ptch2*^{tc294z} mutant environment disrupts cell position, roundness, and orientation angle of transplanted wild type cells, suggesting the presence of non-cell-autonomous factors. A wild type host environment rescues the roundness of *ptch2*^{tc294z} mutant cells, again suggesting the presence of non-cell-autonomous factors. This same wild type environment, however, is not sufficient to convert final position and orientation angle of the mutant cells to that of wild type, suggesting the presence of cell-intrinsic factors regulating these specific phenotypes. We conclude, then, that optic fissure and stalk cell position, roundness, and orientation angle are governed by a combination of cell-autonomous and non-cell-autonomous factors.

Hh transcriptional activity is expanded at early stages in the ptch2^{tc294z} *mutant*

We next set out to determine the signaling mechanisms by which the *ptch2*^{tc294z} mutation disrupts cell motility. Downstream target gene expression is upregulated in the *ptch2*^{tc294z} mutant (Koudijs et al., 2008; Lee et al., 2008), yet this was not examined at early stages of optic cup morphogenesis relevant to when we observe disrupted cell motility. Determining, with cellular resolution, the location(s) of cells exhibiting overactive Hh signaling would help identify the primary source(s) of defects in mutants.

To determine if Hh signaling is increased at early stages of optic cup morphogenesis, we established a new transgenic Hedgehog signaling reporter line, Tg(*GBS-ptch2*:H2A-mCherry)^{z206}, in which a *ptch2* promoter fragment containing an additional Gli recognition sequence (Shen et al., 2013) drives H2A-mCherry. We also generated a new transgenic reporter line marking Shh-producing cells, Tg(*Shh*:H2A-GFP)^{z205}, in which the *Shh*

promoter and arABC downstream enhancer elements drive H2A-GFP (Chang et al., 1997; Muller et al., 1999; Ertzer et al., 2007). With these two lines, we generated double transgenic embryos in which both Shh-producing and Hh-responding cells are marked simultaneously.

At early optic cup morphogenesis stages (12 and 14 hpf), the Hh signaling reporter is significantly expanded along the anterior-posterior axis in *ptch2*^{tc294z} mutant embryos compared to wild type (Figure 6A,B,D,E,G). Expansion persists through the end of optic cup formation (24 hpf; Figure 6C,F). This suggests that Hh signaling in the *ptch2*^{tc294z} mutant is significantly expanded at timepoints relevant to the disrupted cell movements.

In addition, reporter expression also appears increased in intensity. To quantify reporter expression, we used a region of interest capturing the anterior-posterior length of the optic vesicle, and measured fluorescence intensity (Figure 6H-J). We normalized fluorescence intensity in each embryo by dividing the anterior quarter by the posterior quarter (where specific reporter expression was not observed); the reporter is more strongly induced in *ptch2*^{tc294z} mutants compared to wild type embryos (Figure 6J).

To determine whether these results reflect changes in signaling or simply fluorescent reporter protein perdurance, we performed antibody staining for an endogenous target of Hh signaling, Pax2a. We find that the domain of endogenous Pax2a localization is significantly expanded along the anterior-posterior axis of 12 and 14 hpf optic vesicles of *ptch2*^{tc294z} mutants compared to wild type (Figure 6K,L,N,O,Q); expansion persists to optic cup stage (Figure 6M,P). These data indicate that loss of *ptch2* results in significantly expanded and increased Hh transcriptional output during early optic cup morphogenesis, which may play a role in the mechanism by which cell movements are disrupted.

Overactive Hh signaling acts through a canonical Gli1-dependent pathway to disrupt optic fissure and stalk development

We have shown that overactive Hh signaling in the *ptch2*^{tc294z} mutant can lead to increased downstream gene expression. Hh signaling, acting through Gli-dependent transcriptional output, is known as canonical Hh signaling. Additionally, Hh signaling in neuronal growth cones can act more directly on the cytoskeleton, in a transcription-independent manner, through Src-family kinases (SFKs) (Yam et al., 2009). Since disrupted cell motility is a key feature underlying morphogenetic defects, we asked which downstream signaling mechanisms are necessary for *ptch2*^{tc294z} mutation to cause coloboma.

To test a role for non-canonical Hh signaling via SFKs, we utilized two structurally unrelated SFK inhibitors, PP2 and SU6656 (Molina et al., 2007; Murphy et al., 2011; Gallardo et al., 2015). We treated embryos from 5.5-13 hpf, as previously established for rescue of the *ptch2*^{tc294z} mutant coloboma with cyclopamine (Lee et al., 2008). The coloboma phenotype is incompletely penetrant at 52 hpf, therefore, the percent of embryos displaying coloboma was normalized to DMSO control within each individual experiment. As a positive control, we treated embryos with the Smo inhibitor BMS-833923 (Armstrong et al., 2017), which yields ~50% rescue (Figure 7A). In contrast, neither SFK inhibitor rescues the coloboma phenotype (PP2 exacerbated it). We conclude that non-canonical Hh signaling via SFKs is not required to cause coloboma in the *ptch2*^{tc294z} mutant.

To test a role for Gli transcription factors, we carried out a genetic epistasis experiment in which the *ptch2*^{tc294z} mutant was crossed to the *gli1*^{ts269} loss of function mutant (Brand et al., 1996; Karlstrom et al., 1996). Gli1 is the major transcriptional activator downstream of Hedgehog signaling in zebrafish (Karlstrom et al., 2003), and we reasoned that if Gli1 activity is necessary for coloboma, loss of *gli1* would rescue this

phenotype. For these experiments, coloboma was scored at 52-55 hpf (Lee et al., 2008). We find that indeed, loss of one or both copies of *gli1* partially rescues the coloboma phenotype in *ptch2*^{tc294z} mutants (Figure 7B-F). Although it was initially surprising that loss of one copy of *gli1* might confer rescue, haploinsufficient phenotypes have been described with this specific line; *gli1*^{ts269} heterozygotes can display phenotypes (Tyurina et al., 2005). This rescue is incomplete when removing either one or both copies of Gli1 (Figure 7B), suggesting other contributions, possibly from other Gli proteins (such as Gli2 or Gli3), or other Hh signaling outputs. We conclude that overactive Hh signaling acts at least in part via canonical Gli1 activity to disrupt eye development.

Discussion

We describe here, for the first time, the cell movements underlying optic fissure and stalk formation. Somewhat surprisingly, we find that neighboring cells contribute to opposing margins of the optic fissure. This suggests that optic fissure formation may not be the passive byproduct of optic cup invagination: if the fissure margins were simply folded together, we might expect opposing margins to be derived from distant positions, with cells taking reversed trajectories to abut each other on either side of the fissure. In contrast, we see that neighboring cell populations take similar trajectories but come to reside on opposite margins. This suggests that there may be an active process, potentially involving tissue involution or folding, to split these cell populations. Future work will address the mechanisms underlying this process.

Uncovering the cell movements underlying optic fissure and stalk formation allowed us to establish a framework to identify cell migration and motility defects that might contribute to impaired development and coloboma. We then determined the defects in cell movements underlying optic fissure and stalk formation in the *ptch2*^{tc294z} mutant

(Figure 3). By performing 4D cell tracking, we detected motility defects in cells that normally contribute to optic fissure and optic stalk, but in the mutant, fail to achieve their correct position due to defective motility. Cells that should contribute to optic fissure and cup instead contribute to the optic stalk. Temporal fissure cells in the *ptch2*^{tc294z} mutant arise from a novel location, which, under normal conditions, contributes to temporal retina (Figure S1). At 12 hpf, this novel origin does not exhibit increased Hh target gene expression in *ptch2*^{tc294z} mutants (Figure 6), therefore, we hypothesize that the novel trajectory observed is secondary to failure of midline cells to properly move into the optic stalk and optic cup.

Our results indicate that cell motility can be affected by overactive Hh signaling; we sought to understand how this might occur. Hedgehog signaling can act through both canonical Gli-dependent and non-canonical Gli-independent pathways; we find evidence for Gli1-dependent activity but not non-canonical SFK signaling in the etiology of the *ptch2*^{tc294z} mutant coloboma phenotype. Using cell transplantation, we also find that the *ptch2*^{tc294z} mutation acts via both cell-autonomous and non-cell-autonomous mechanisms to regulate cell position and orientation, but non-cell-autonomous mechanisms largely regulate cell elongation. Taken together, we propose the following model (Figure 8): in a wild type embryo, the nasal and temporal optic fissure margins and optic stalk are derived from neighboring cell populations. The cells move in a wide J-shaped trajectory, with temporal cells moving first, followed by nasal cells, followed by stalk cells (Figure 8A). In the *ptch2*^{tc294z} mutant, cells derived from the prospective optic fissure and stalk domains do not achieve their correct positions (Figure 8B) and have aberrant morphology: they are significantly less elongated, and more variably oriented with respect to the anterior-posterior axis of the embryo. Although cell morphology is altered, tissue polarity is still intact. We speculate that optic fissure and stalk formation

fail because inappropriate positioning of prospective fissure and stalk cells impedes formation of a functional structure.

How might overactive Hh signaling actually affect cell motility? In our working model, we propose that in a wild type embryo, Hedgehog signaling leads to Gli1-dependent induction of crucial factors that control cell elongation, orientation, and motility. At least one factor, “Factor X” (Figure 8C, *green*) acts in a non-cell autonomous manner, potentially as a cell surface or secreted molecule. We hypothesize that expression level of Factor X is critical for regulating motile cell behaviors, either directly or indirectly, and that in the *ptch2*^{tc294z} mutant, expansion and overexpression of Factor X leads to altered cell elongation and orientation, and disruption of motility (Figure 8D). We are currently working to identify Factor X.

Our results reveal the cellular basis of the optic fissure and stalk defect in one coloboma model: moving forward, and using our wild type analysis as a framework for comparison, it will be important to examine other coloboma models with aberrant optic fissure and stalk formation. We speculate that there may be many ways to disrupt this process, and multiple cell populations that can be affected, reflecting the genetic heterogeneity found in the human condition.

Author Contributions

H.B.G., S.L., and K.M.K. were responsible for conceptualization, methodology, and investigation through the study. K.R.C. developed the Abacus 3D-cell-counting MATLAB script. E.O.W. and B.F.M. generated key resources and methodology. K.M.K., H.B.G., and S.L. were responsible for visualization, writing, and editing of the manuscript.

Acknowledgments

We are grateful to Jeff Gross, Rolf Karlstrom, Alex Schier, Uwe Strähle, Ferenc Müller, and Yavor Hadziev for generously providing reagents. Thanks to Mick Jurynec and David Grunwald for technical assistance and support, and Kryn Stankunas for advice. Thanks to members of the Kwan lab for useful discussions, and Charlie Murtaugh for critical reading of the manuscript. Thanks to the Centralized Zebrafish Animal Resource and the Mutation Generation and Detection Core at the University of Utah. This work was supported by grants to K.M.K. from the Knights Templar Eye Foundation, a March of Dimes Basil O'Connor Starter Scholar Award, and the NEI/NIH (R01 EY025378, R01 EY025780).

References

Adler, R. and Canto-Soler, M. V. (2007). Molecular mechanisms of optic vesicle development: complexities, ambiguities and controversies. *Dev Biol* **305**, 1-13.

Ando, R., Hama, H., Yamamoto-Hino, M., Mizuno, H. and Miyawaki, A. (2002). An optical marker based on the UV-induced green-to-red photoconversion of a fluorescent protein. *Proc Natl Acad Sci U S A* **99**, 12651-12656.

Armstrong, B. E., Henner, A., Stewart, S. and Stankunas, K. (2017). Shh promotes direct interactions between epidermal cells and osteoblast progenitors to shape regenerated zebrafish bone. *Development* **144**, 1165-1176.

Asai-Coakwell, M., French, C. R., Berry, K. M., Ye, M., Koss, R., Somerville, M., Mueller, R., van Heyningen, V., Waskiewicz, A. J. and Lehmann, O. J. (2007). GDF6, a novel locus for a spectrum of ocular developmental anomalies. *Am J Hum Genet* **80**, 306-315.

Azuma, N., Yamaguchi, Y., Handa, H., Tadokoro, K., Asaka, A., Kawase, E. and Yamada, M. (2003). Mutations of the PAX6 gene detected in patients with a variety of optic-nerve malformations. *Am J Hum Genet* **72**, 1565-1570.

Baulmann, D. C., Ohlmann, A., Flugel-Koch, C., Goswami, S., Cvekl, A. and Tamm, E. R. (2002). Pax6 heterozygous eyes show defects in chamber angle differentiation that are associated with a wide spectrum of other anterior eye segment abnormalities. *Mech Dev* **118**, 3-17.

Bazin-Lopez, N., Valdivia, L. E., Wilson, S. W. and Gestri, G. (2015). Watching eyes take shape. *Curr Opin Genet Dev* **32**, 73-79.

Bernstein, C. S., Anderson, M. T., Gohel, C., Slater, K., Gross, J. M. and Agarwala, S. (2018). The cellular bases of choroid fissure formation and closure. *Dev Biol* **440**, 137-151.

Brand, M., Heisenberg, C. P., Warga, R. M., Pelegri, F., Karlstrom, R. O., Beuchle, D., Picker, A., Jiang, Y. J., Furutani-Seiki, M., van Eeden, F. J. et al. (1996). Mutations affecting development of the midline and general body shape during zebrafish embryogenesis. *Development* **123**, 129-142.

Bryan, C. D., Chien, C. B. and Kwan, K. M. (2016). Loss of laminin alpha 1 results in multiple structural defects and divergent effects on adhesion during vertebrate optic cup morphogenesis. *Dev Biol* **416**, 324-337.

Chang, B. E., Blader, P., Fischer, N., Ingham, P. W. and Strahle, U. (1997). Axial (HNF3beta) and retinoic acid receptors are regulators of the zebrafish sonic hedgehog promoter. *EMBO J* **16**, 3955-3964.

Chow, R. L. and Lang, R. A. (2001). Early eye development in vertebrates. *Annu Rev Cell Dev Biol* **17**, 255-296.

Eccles, M. R. and Schimmenti, L. A. (1999). Renal-coloboma syndrome: a multi-system developmental disorder caused by PAX2 mutations. *Clin Genet* **56**, 1-9.

Ertzer, R., Muller, F., Hadzhiev, Y., Rathnam, S., Fischer, N., Rastegar, S. and Strahle, U. (2007). Cooperation of sonic hedgehog enhancers in midline expression. *Dev Biol* **301**, 578-589.

Evangelista, M., Tian, H. and de Sauvage, F. J. (2006). The hedgehog signaling pathway in cancer. *Clin Cancer Res* **12**, 5924-5928.

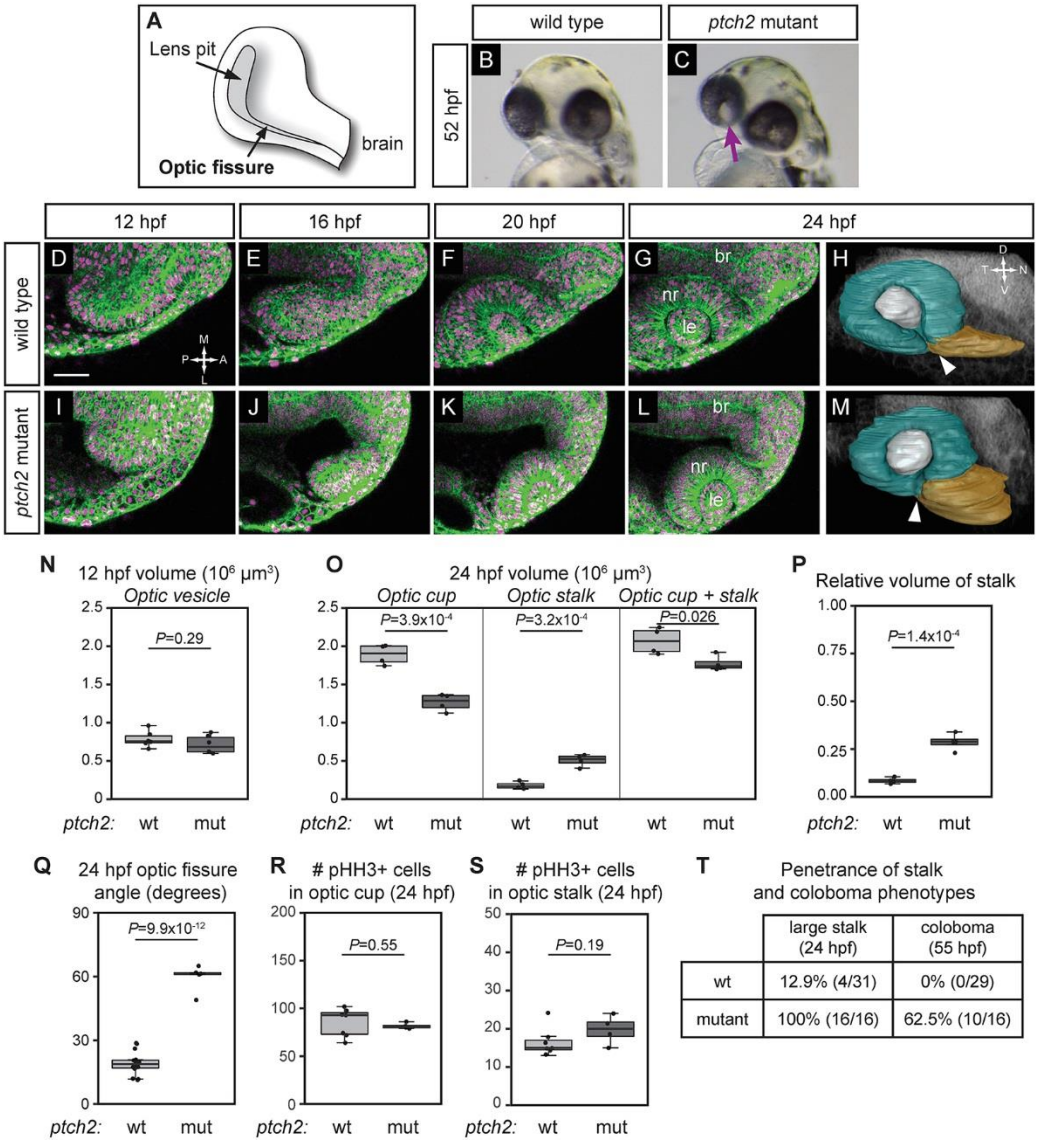
Favor, J., Sandulache, R., Neuhauser-Klaus, A., Pretsch, W., Chatterjee, B., Senft, E., Wurst, W., Blanquet, V., Grimes, P., Sporle, R. et al. (1996). The mouse Pax2(1Neu) mutation is identical to a human PAX2 mutation in a family with renal-coloboma syndrome and results in developmental defects of the brain, ear, eye, and kidney. *Proc Natl Acad Sci U S A* **93**, 13870-13875.

Fitzpatrick, D. R. and van Heyningen, V. (2005). Developmental eye disorders. *Curr Opin Genet Dev* **15**, 348-353.
French, C. R., Stach, T. R., March, L. D., Lehmann, O. J. and Waskiewicz, A. J. (2013). Apoptotic and proliferative defects characterize ocular development in a microphthalmic BMP model. *Invest Ophthalmol Vis Sci* **54**, 4636-4647.
Fuhrmann, S. (2010). Eye morphogenesis and patterning of the optic vesicle. *Curr Top Dev Biol* **93**, 61-84.
Gage, P. J., Suh, H. and Camper, S. A. (1999). Dosage requirement of Pitx2 for development of multiple organs. *Development* **126**, 4643-4651.
Gallardo, V. E., Varshney, G. K., Lee, M., Bupp, S., Xu, L., Shinn, P., Crawford, N. P., Inglese, J. and Burgess, S. M. (2015). Phenotype-driven chemical screening in zebrafish for compounds that inhibit collective cell migration identifies multiple pathways potentially involved in metastatic invasion. *Dis Model Mech* **8**, 565-576.
Gregory-Evans, C. Y., Williams, M. J., Halford, S. and Gregory-Evans, K. (2004). Ocular coloboma: a reassessment in the age of molecular neuroscience. *J Med Genet* **41**, 881-891.
Hahn, H., Wicking, C., Zaphiropoulous, P. G., Gailani, M. R., Shanley, S., Chidambaram, A., Vorechovsky, I., Holmberg, E., Unden, A. B., Gillies, S. et al. (1996). Mutations of the human homolog of Drosophila patched in the nevoid basal cell carcinoma syndrome. *Cell* **85**, 841-851.
Heermann, S., Schutz, L., Lemke, S., Krieglstein, K. and Wittbrodt, J. (2015). Eye morphogenesis driven by epithelial flow into the optic cup facilitated by modulation of bone morphogenetic protein. *Elife* **4**.
Hero, I. (1989). The optic fissure in the normal and microphthalmic mouse. *Exp Eye Res* **49**, 229-239.
Hero, I. (1990). Optic fissure closure in the normal cinnamon mouse. An ultrastructural study. *Invest Ophthalmol Vis Sci* **31**, 197-216.
Hero, I., Farjah, M. and Scholtz, C. L. (1991). The prenatal development of the optic fissure in colobomatous microphthalmia. *Invest Ophthalmol Vis Sci* **32**, 2622-2635.
Huang, P. and Schier, A. F. (2009). Dampened Hedgehog signaling but normal Wnt signaling in zebrafish without cilia. *Development* **136**, 3089-3098.
James, A., Lee, C., Williams, A. M., Angileri, K., Lathrop, K. L. and Gross, J. M. (2016). The hyaloid vasculature facilitates basement membrane breakdown during choroid fissure closure in the zebrafish eye. *Dev Biol* **419**, 262-272.
Karlstrom, R. O., Tyurina, O. V., Kawakami, A., Nishioka, N., Talbot, W. S., Sasaki, H. and Schier, A. F. (2003). Genetic analysis of zebrafish gli1 and gli2 reveals divergent requirements for gli genes in vertebrate development. *Development* **130**, 1549-1564.
Karlstrom, R. O., Trowe, T., Klostermann, S., Baier, H., Brand, M., Crawford, A. D., Grunewald, B., Haffter, P., Hoffmann, H., Meyer, S. U. et al. (1996). Zebrafish mutations affecting retinotectal axon pathfinding. *Development* **123**, 427-438.
Keller, S. A., Jones, J. M., Boyle, A., Barrow, L. L., Killen, P. D., Green, D. G., Kapousta, N. V., Hitchcock, P. F., Swank, R. T. and Meisler, M. H. (1994). Kidney and retinal defects (Krd), a transgene-induced mutation with a deletion of mouse chromosome 19 that includes the Pax2 locus. *Genomics* **23**, 309-320.
Kim, T. H., Goodman, J., Anderson, K. V. and Niswander, L. (2007). Phactr4 regulates neural tube and optic fissure closure by controlling PP1-, Rb-, and E2F1-regulated cell-cycle progression. *Dev Cell* **13**, 87-102.
Kimmel, C. B., Ballard, W. W., Kimmel, S. R., Ullmann, B. and Schilling, T. F. (1995). Stages of embryonic development of the zebrafish. *Dev Dyn* **203**, 253-310.
Konieczny, A. and Ausubel, F. M. (1993). A procedure for mapping Arabidopsis mutations using co-dominant ecotype-specific PCR-based markers. *Plant J* **4**, 403-410.

Koudijs, M. J., den Broeder, M. J., Groot, E. and van Eeden, F. J. (2008). Genetic analysis of the two zebrafish patched homologues identifies novel roles for the hedgehog signaling pathway. *BMC Dev Biol* **8**, 15.
Kwan, K. M., Otsuna, H., Kidokoro, H., Carney, K. R., Saijoh, Y. and Chien, C. B. (2012). A complex choreography of cell movements shapes the vertebrate eye. *Development* **139**, 359-372.
Kwan, K. M., Fujimoto, E., Grabher, C., Mangum, B. D., Hardy, M. E., Campbell, D. S., Parant, J. M., Yost, H. J., Kanki, J. P. and Chien, C. B. (2007). The Tol2kit: a multisite gateway-based construction kit for Tol2 transposon transgenesis constructs. *Dev Dyn* **236**, 3088-3099.
Lee, J., Willer, J. R., Willer, G. B., Smith, K., Gregg, R. G. and Gross, J. M. (2008). Zebrafish blowout provides genetic evidence for Patched1-mediated negative regulation of Hedgehog signaling within the proximal optic vesicle of the vertebrate eye. *Dev Biol* **319**, 10-22.
Lupo, G., Gestri, G., O'Brien, M., Denton, R. M., Chandraratna, R. A., Ley, S. V., Harris, W. A. and Wilson, S. W. (2011). Retinoic acid receptor signaling regulates choroid fissure closure through independent mechanisms in the ventral optic cup and periocular mesenchyme. *Proc Natl Acad Sci U S A* **108**, 8698-8703.
Macdonald, R., Scholes, J., Strahle, U., Brennan, C., Holder, N., Brand, M. and Wilson, S. W. (1997). The Pax protein Noi is required for commissural axon pathway formation in the rostral forebrain. *Development* **124**, 2397-2408.
Martinez-Morales, J. R. and Wittbrodt, J. (2009). Shaping the vertebrate eye. *Curr Opin Genet Dev* **19**, 511-517.
Molina, G. A., Watkins, S. C. and Tsang, M. (2007). Generation of FGF reporter transgenic zebrafish and their utility in chemical screens. *BMC Dev Biol* **7**, 62.
Muller, F., Chang, B., Albert, S., Fischer, N., Tora, L. and Strahle, U. (1999). Intronic enhancers control expression of zebrafish sonic hedgehog in floor plate and notochord. *Development* **126**, 2103-2116.
Murphy, D. A., Diaz, B., Bromann, P. A., Tsai, J. H., Kawakami, Y., Maurer, J., Stewart, R. A., Izpisua-Belmonte, J. C. and Courtneidge, S. A. (2011). A Src-Tks5 pathway is required for neural crest cell migration during embryonic development. *PLoS One* **6**, e22499.
Neff, M. M., Neff, J. D., Chory, J. and Pepper, A. E. (1998). dCAPS, a simple technique for the genetic analysis of single nucleotide polymorphisms: experimental applications in Arabidopsis thaliana genetics. *Plant J* **14**, 387-392.
Ozeki, H., Shirai, S., Ikeda, K. and Ogura, Y. (1999). Anomalies associated with Axenfeld-Rieger syndrome. *Graefes Arch Clin Exp Ophthalmol* **237**, 730-734.
Parant, J. M., George, S. A., Pryor, R., Wittwer, C. T. and Yost, H. J. (2009). A rapid and efficient method of genotyping zebrafish mutants. *Dev Dyn* **238**, 3168-3174.
Patel, A. and Sowden, J. C. (2017). Genes and pathways in optic fissure closure. *Semin Cell Dev Biol*.
Picker, A., Cavodeassi, F., Machate, A., Bernauer, S., Hans, S., Abe, G., Kawakami, K., Wilson, S. W. and Brand, M. (2009). Dynamic coupling of pattern formation and morphogenesis in the developing vertebrate retina. *PLoS Biol* **7**, e1000214.
Pillai-Kastoori, L., Wen, W., Wilson, S. G., Strachan, E., Lo-Castro, A., Fichera, M., Musumeci, S. A., Lehmann, O. J. and Morris, A. C. (2014). Sox11 is required to maintain proper levels of Hedgehog signaling during vertebrate ocular morphogenesis. *PLoS Genet* **10**, e1004491.
Ragge, N. K., Salt, A., Collin, J. R., Michalski, A. and Farndon, P. A. (2005). Gorlin syndrome: the PTCH gene links ocular developmental defects and tumour formation. *Br J Ophthalmol* **89**, 988-991.

Sanyanusin, P., Schimmenti, L. A., McNoe, L. A., Ward, T. A., Pierpont, M. E., Sullivan, M. J., Dobyns, W. B. and Eccles, M. R. (1995). Mutation of the PAX2 gene in a family with optic nerve colobomas, renal anomalies and vesicoureteral reflux. *Nat Genet* **9**, 358-364.
Schimmenti, L. A., de la Cruz, J., Lewis, R. A., Karkera, J. D., Manligas, G. S., Roessler, E. and Muenke, M. (2003). Novel mutation in sonic hedgehog in non-syndromic colobomatous microphthalmia. *Am J Med Genet A* **116A**, 215-221.
Sepich, D. S., Usmani, M., Pawlicki, S. and Solnica-Krezel, L. (2011). Wnt/PCP signaling controls intracellular position of MTOCs during gastrulation convergence and extension movements. *Development* **138**, 543-552.
Shen, M. C., Ozacar, A. T., Osgood, M., Boeras, C., Pink, J., Thomas, J., Kohtz, J. D. and Karlstrom, R. (2013). Heat-shock-mediated conditional regulation of hedgehog/gli signaling in zebrafish. *Dev Dyn* **242**, 539-549.
Sidhaye, J. and Norden, C. (2017). Concerted action of neuroepithelial basal shrinkage and active epithelial migration ensures efficient optic cup morphogenesis. *Elife* **6**.
Singh, S., Mishra, R., Arango, N. A., Deng, J. M., Behringer, R. R. and Saunders, G. F. (2002). Iris hypoplasia in mice that lack the alternatively spliced Pax6(5a) isoform. *Proc Natl Acad Sci U S A* **99**, 6812-6815.
Sinn, R. and Wittbrodt, J. (2013). An eye on eye development. *Mech Dev* **130**, 347-358.
Smyth, I., Narang, M. A., Evans, T., Heimann, C., Nakamura, Y., Chenevix-Trench, G., Pietsch, T., Wicking, C. and Wainwright, B. J. (1999). Isolation and characterization of human patched 2 (PTCH2), a putative tumour suppressor gene inbasal cell carcinoma and medulloblastoma on chromosome 1p32. *Hum Mol Genet* **8**, 291-297.
Stull, D. L. and Wikler, K. C. (2000). Retinoid-dependent gene expression regulates early morphological events in the development of the murine retina. *J Comp Neurol* **417**, 289-298.
Torres, M., Gomez-Pardo, E. and Gruss, P. (1996). Pax2 contributes to inner ear patterning and optic nerve trajectory. *Development* **122**, 3381-3391.
Tyurina, O. V., Guner, B., Popova, E., Feng, J., Schier, A. F., Kohtz, J. D. and Karlstrom, R. O. (2005). Zebrafish Gli3 functions as both an activator and a repressor in Hedgehog signaling. *Dev Biol* **277**, 537-556.
Wan, Y., Otsuna, H., Chien, C. B. and Hansen, C. (2012). FluoRender: An Application of 2D Image Space Methods for 3D and 4D Confocal Microscopy Data Visualization in Neurobiology Research. *IEEE Pac Vis Symp*, 201-208.
Wen, W., Pillai-Kastoori, L., Wilson, S. G. and Morris, A. C. (2015). Sox4 regulates choroid fissure closure by limiting Hedgehog signaling during ocular morphogenesis. *Dev Biol* **399**, 139-153.
Williamson, K. A. and FitzPatrick, D. R. (2014). The genetic architecture of microphthalmia, anophthalmia and coloboma. *Eur J Med Genet* **57**, 369-380.
Yam, P. T., Langlois, S. D., Morin, S. and Charron, F. (2009). Sonic hedgehog guides axons through a noncanonical, Src-family-kinase-dependent signaling pathway. *Neuron* **62**, 349-362.
Yang, X. J. (2004). Roles of cell-extrinsic growth factors in vertebrate eye pattern formation and retinogenesis. *Semin Cell Dev Biol* **15**, 91-103.
Zhou, C. J., Molotkov, A., Song, L., Li, Y., Pleasure, D. E., Pleasure, S. J. and Wang, Y. Z. (2008). Ocular coloboma and dorsoventral neuroretinal patterning defects in Lrp6 mutant eyes. *Dev Dyn* **237**, 3681-3689.

Figures



	large stalk (24 hpf)	coloboma (55 hpf)
wt	12.9% (4/31)	0% (0/29)
mutant	100% (16/16)	62.5% (10/16)

Figure 1. Optic cup morphogenesis and optic fissure and stalk formation are disrupted in the *ptch2*^{tc294z} mutant.
(A) Schematic of optic fissure at optic cup stage, 24 hpf.
(B) Wild type embryo, 52 hpf: the eye is evenly pigmented.
(C) *ptch2*^{tc294z} mutant embryo, 52 hpf: coloboma is apparent as a region of hypopigmentation in the eye (*arrow*).
(D-G) Wild type optic cup formation, single confocal slices from 4-dimensional imaging data set (12-24 hpf). Dorsal view. EGFP-CAAX (membranes, *green*), H2A.F/Z-mCherry (nuclei, *magenta*).

(H) Volume rendering of wild type embryo, 24 hpf. Lateral view. Optic cup (*teal*), lens (*gray*), optic stalk (*gold*). *Arrowhead*, optic fissure.
(I-L) *ptch2*^{tc294z} mutant optic cup formation. Dorsal views. EGFP-CAAX (membranes, *green*), H2A.F/Z-mCherry (nuclei, *magenta*).
(M) Volume rendering of *ptch2*^{tc294z} mutant embryo, 24 hpf. Lateral view. Optic cup (*teal*), lens (*gray*), optic stalk (*gold*). *Arrowhead*, optic fissure, which has not formed correctly.
(N) Optic vesicle volume, 12 hpf.
(O) Optic cup, stalk, and total volume, 24 hpf.
(P) Relative volume of stalk as a proportion of optic cup+stalk volume.
(Q) Angle measurement of optic fissure opening, 24 hpf.
(R) Number of phospho-histone H3-positive cells in the optic cup, 24 hpf.
(S) Number of phospho-histone H3-positive cells in the optic stalk, 24 hpf.
(T) Comparison of penetrance of large stalk and coloboma phenotypes at 24 and 48 hpf, respectively.
(N,O,Q,S) *P* values, unpaired student's t-test. (P,R) *P* values, unpaired Welch's t-test to account for unequal variance.
br, brain; *nr*, neural retina; *le*, lens. RPE is very flattened and difficult to see; thus it is not labeled. Scale bar, 50 µm.

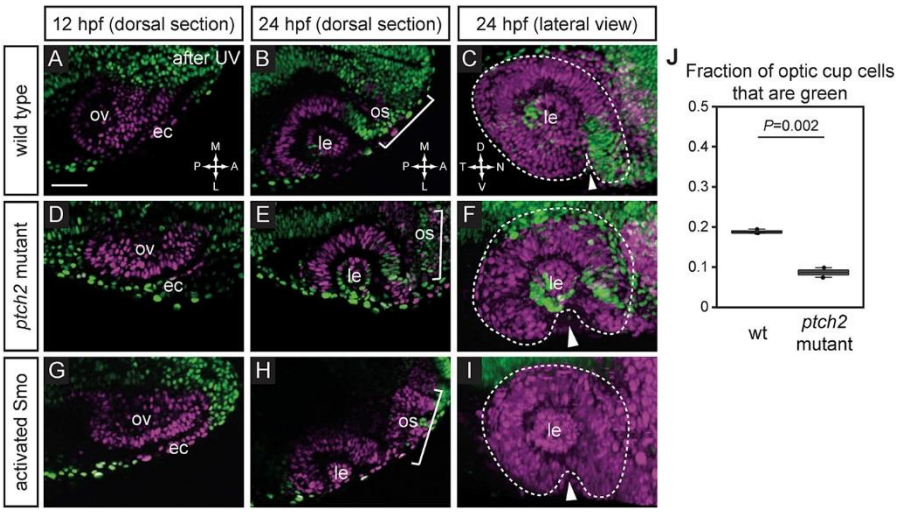


Figure 2. Extended evagination cell movements are disrupted by overactive Hh signaling.

(A-C) Wild type; (D-F) *ptch2*^{tc294z} mutant; or (G-I) activated Smo RNA injected embryos subjected to nls-Kaede photoconversion at optic vesicle stage (A,D,G). The entire optic vesicle was converted from green to red (*magenta*) fluorescence (along with some ectoderm). (B,E,H) Single confocal sections of optic cups from photoconverted embryos, 24 hpf, dorsal view. (C,F,I) 3D-rendered optic cups, 24 hpf, lateral view. Dashed line outlines the optic cup. Arrowhead, optic fissure.

(J) Quantification of extended evagination movements: proportion of green nuclei in the optic cup (green nuclei divided by the total number of nuclei). The proportion of green nuclei in the optic cup is significantly reduced in *ptch2*^{tc294z} mutants. *P* value, unpaired student's t-test.

ov, optic vesicle; *ec*, ectoderm; *le*, lens; *os*, optic stalk. Scale bar, 50 μm.

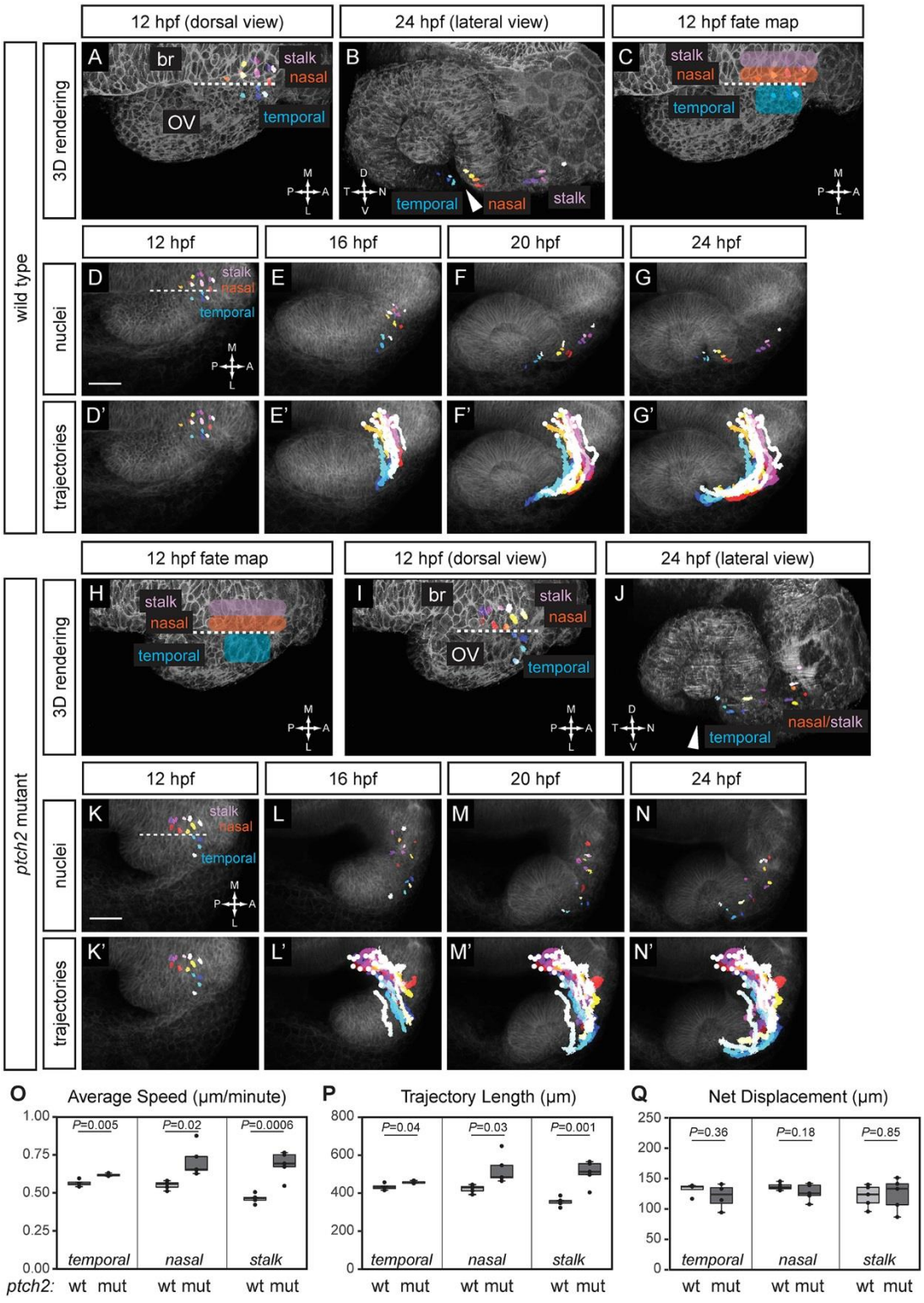


Figure 3. Origins and trajectories of cells contributing to the optic fissure are disrupted in the *ptch2*^{tc294z} mutant.
(A-G') Wild type optic fissure and stalk cell movements (12-24 hpf). (A) Rendering of nuclei and membrane channel, 12 hpf, dorsal view. (B) Rendering of nuclei and membrane channel, 24 hpf, lateral view. (C) Fate map of optic fissure and stalk cells. (D-G) Nuclei over membrane channel average projection. (D'-G') Trajectories over membrane channel average projection.
(H-N') *ptch2*^{tc294z} mutant optic fissure cell movements (12-24 hpf). (H) Application of wild type fate map to *ptch2*^{tc294z} mutant optic vesicle. (I) Rendering of selected nuclei and membrane channel, 12 hpf, dorsal view. (J) Rendering of nuclei and membrane channel, 24 hpf, lateral view. (K-N) Nuclei over membrane channel average projection. (K'-N') Trajectories over membrane channel average projection.
(O-Q) Quantification of cell tracking data. (O) 3-dimensional average speed. (P) 3-dimensional trajectory length. (Q) 3-dimensional net displacement.
White circles, trajectory origins. Temporal nuclei, blue shades; nasal nuclei, red/yellow shades; stalk nuclei, purple shades. Dotted line, boundary between optic vesicle and midline region. Scale bar, 50 μm.

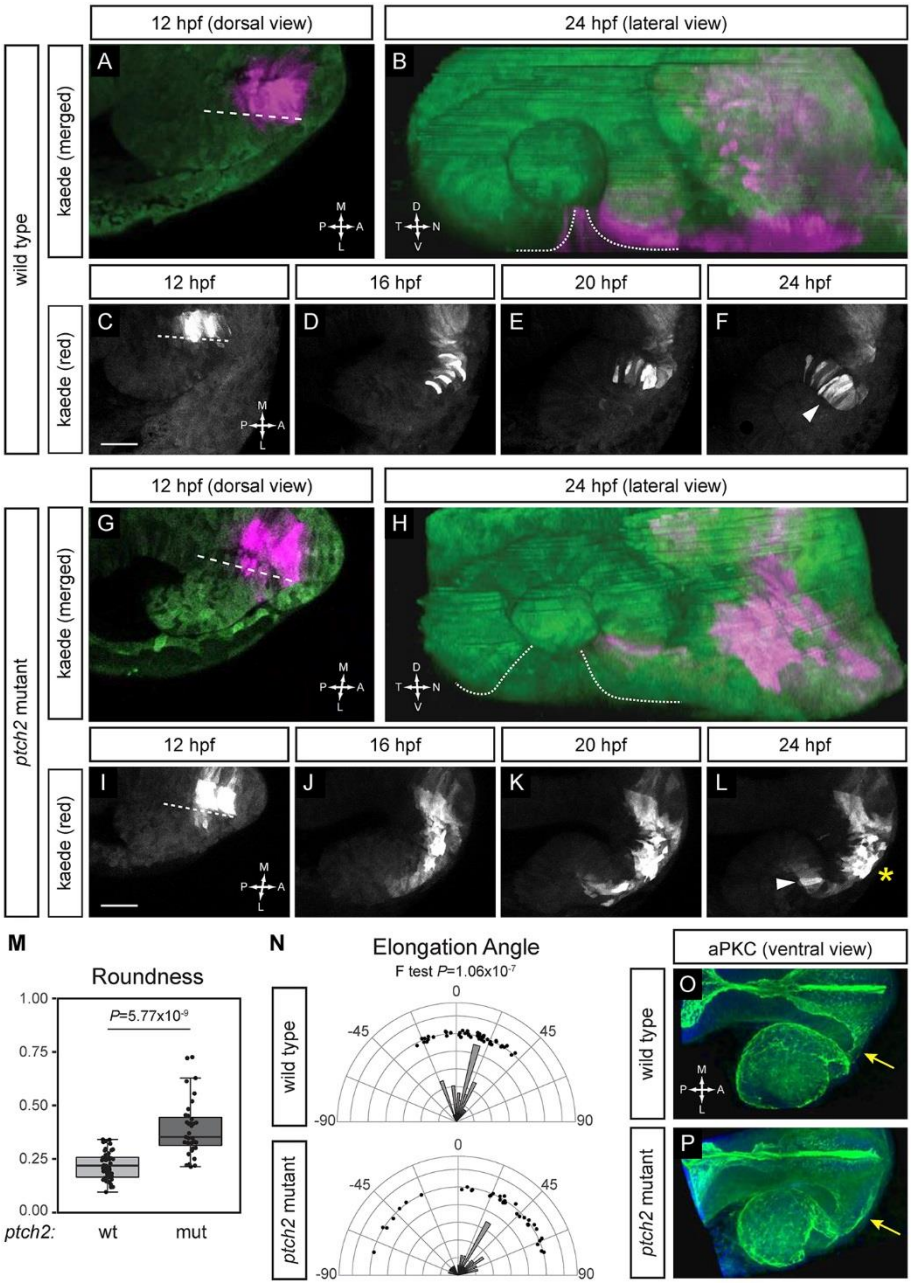


Figure 4. Migratory behaviors of cells contributing to optic fissure formation are disrupted in the *ptch2*^{tc294z} mutant.

(A-F) Wild type cell movements, cells marked with cytoplasmic Kaede. (A) Merged image of marked cells, 12 hpf, dorsal view. (B) Merged rendering of marked cells, 24 hpf, lateral view. (C-F) Projections from 4-dimensional imaging data set of marked cells moving to ventronasal retina and optic fissure nasal margin.

(G-L) Cell movements in the *ptch2*^{tc294z} mutant. (G) Merged image of marked cells, 12 hpf, dorsal view. (H) Merged rendering of marked cells, 24 hpf, lateral view. (I-L) Projections from 4-dimensional imaging data set of marked cells: most cells (asterisk) remain in the optic stalk.

(M) Roundness. *P* value, unpaired Welch's t-test to account for unequal variance.
(N) Elongation angle. *P* value, F-test to determine the probability that the variances are not significantly different.
(O, P) aPKC antibody staining, 24 hpf, ventral view. No gross defects in apicobasal polarity in *ptch2*^{tc294z} mutants are apparent. (O) Wild type. (P) *ptch2*^{tc294z} mutant. Yellow arrows, optic stalk.
Dashed lines, boundary between optic vesicle and midline region. Dotted lines, optic fissure margins. Arrowheads, bipolar cells; asterisk, multipolar cells. Scale bar, 50 µm.

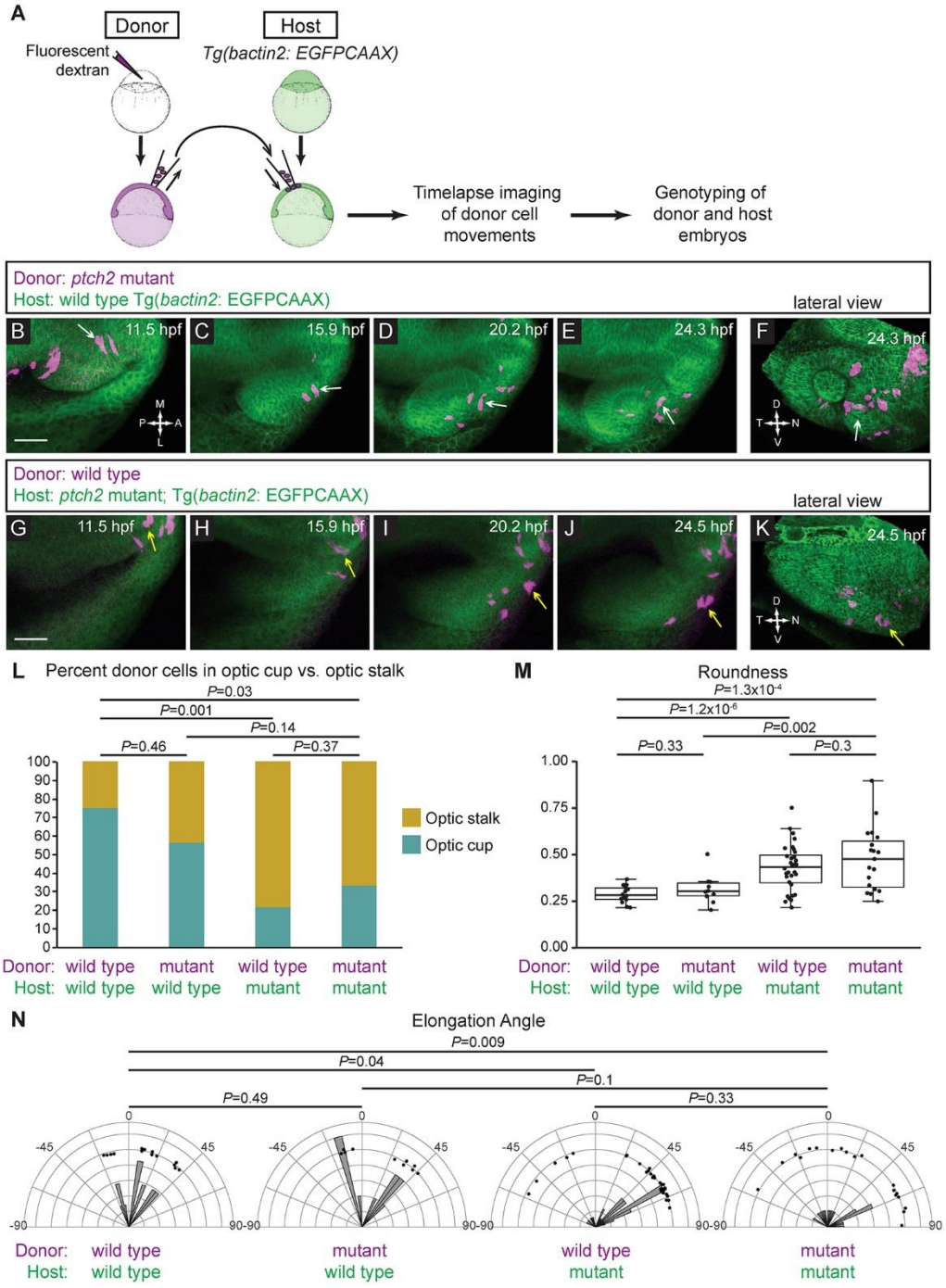


Figure 5. *ptch2*^{tc294z} acts in a non-cell-autonomous manner to disrupt cell movements.

(A) Schematic of cell transplantation approach. Cells were transplanted from donor to host at blastula stages, and timelapse microscopy performed on embryos in which transplanted cells were in the prospective optic fissure forming region at 12 hpf. After imaging, both donor and host embryos were genotyped.

(B-F) Transplantation of *ptch2*^{tc294z} mutant cells into a wild type host. (B-E) Mutant cells exhibit an elongated morphology, move out of the midline region, through the optic stalk,

and into the optic cup, where a mutant donor cell contributes to the optic fissure (*white arrows*). (F) Lateral view of 3D rendering, final timepoint. White arrow marks cell that contributes directly to optic fissure margin.
(G-K) Transplantation of wild type cells into a *ptch2*^{tc294z} mutant host. (G-J) Wild type cells move out of the midline region; few cells move to the central retina within the optic cup. Most cells become less elongated and reside in the optic stalk region (*yellow arrows*). (K) Lateral view of 3D rendering, final timepoint. Yellow arrow marks cells that arose from prospective optic cup/fissure region, but contribute to optic stalk.
(L-N) Quantification of transplant results. (L) Proportions of transplanted cells contributing to optic stalk (*gold*) or optic cup/fissure (*teal*). *P* values, Fisher's exact test. (M) Roundness of transplanted cells: morphology reflects host genotype. *P* value, unpaired Welch's t-test to account for unequal variance. (N) Elongation angle of transplanted cells. *P* value, F-test to determine the probability that the variances are not significantly different.
Scale bar, 50 µm.

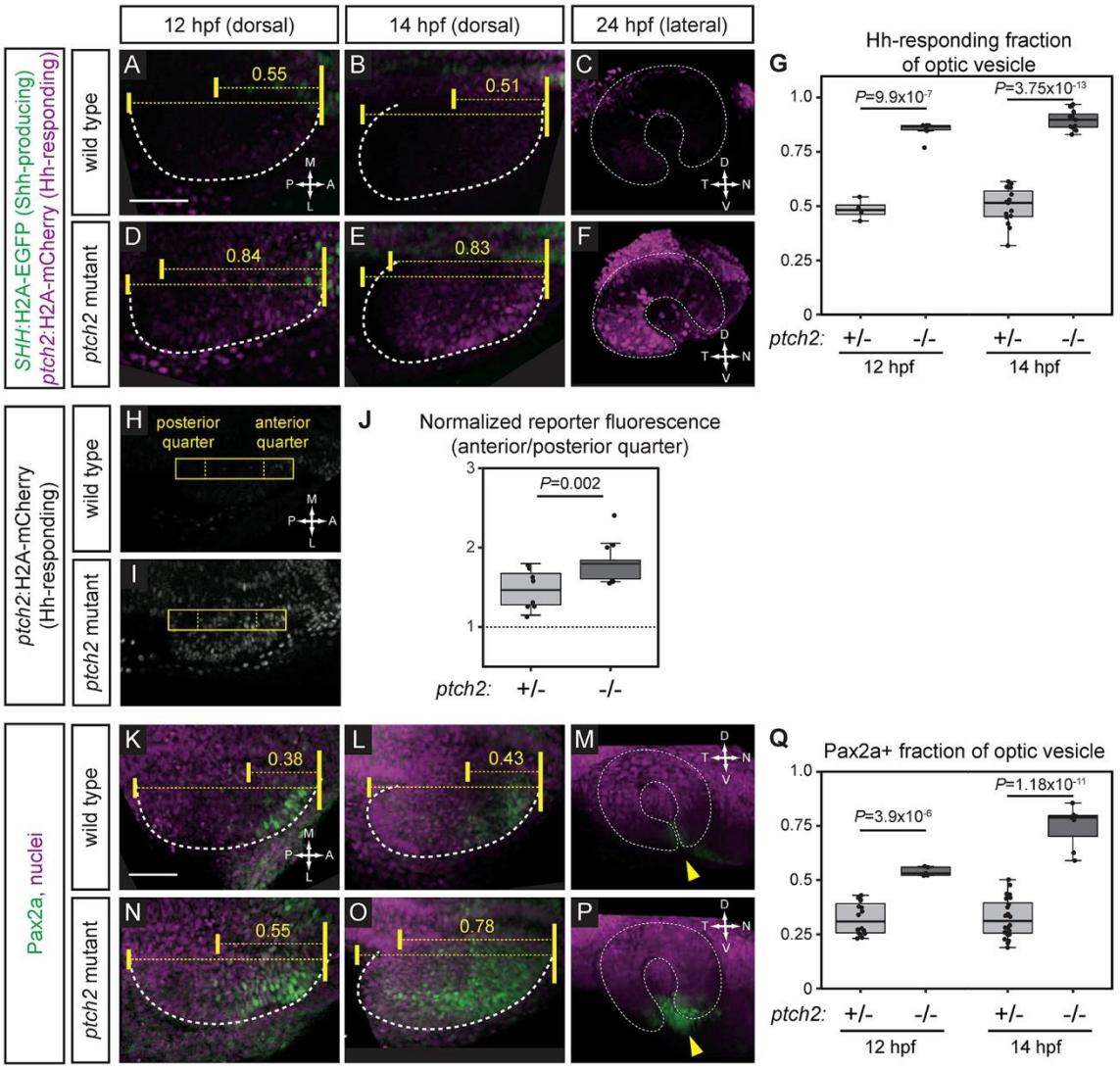


Figure 6. Hedgehog signaling is increased at early optic vesicle stages in the *ptch2*^{tc294z} mutant.

(A-G) Hedgehog signaling reporter expression is expanded in the *ptch2*^{tc294z} mutant at 12, 14, and 24 hpf. Embryos are labeled for Shh-producing cells (*green*; Tg(*SHH*:H2A-GFP)^{z205}) and Hh-responding cells (*magenta*; Tg(*GBS-ptch2*:H2A-mCherry)^{z206}). (A-C) Wild type double transgenic. (D-F) *ptch2*^{tc294z} mutant double transgenic. (G) Quantification of the anterior-posterior fraction of the optic vesicle occupied by reporter-positive cells.

(H-J) Hedgehog signaling reporter expression (Tg(*GBS-ptch2*:H2A-mCherry)^{z206}) is upregulated in the *ptch2*^{tc294z} mutant at 13 hpf. A rectangle (44 pixels wide) spanning the anterior-posterior length of the optic vesicle was drawn. Fluorescence intensity in the anterior quarter was normalized to the posterior quarter. A value of 1 (J, dotted line) represents equal fluorescence intensity in the quarters. (H) Wild type transgenic. (I) *ptch2*^{tc294z} mutant transgenic. (J) Normalized fluorescence intensity.

(K-Q) Endogenous Hedgehog target gene expression (Pax2a) is expanded in the *ptch2*^{tc294z} mutant at 12, 14, and 24 hpf. Antibody staining performed for Pax2a (*green*),

nuclei counterstained with TO-PRO-3 (*magenta*). (K-M) Wild type. (N-P) *ptch2*^{tc294z} mutant. *Yellow arrowheads*, optic fissure. (Q) Quantification of the anterior-posterior fraction of optic vesicle occupied by Pax2a-positive cells.
P-values, unpaired student's t-tests. Scale bar, 50 µm.

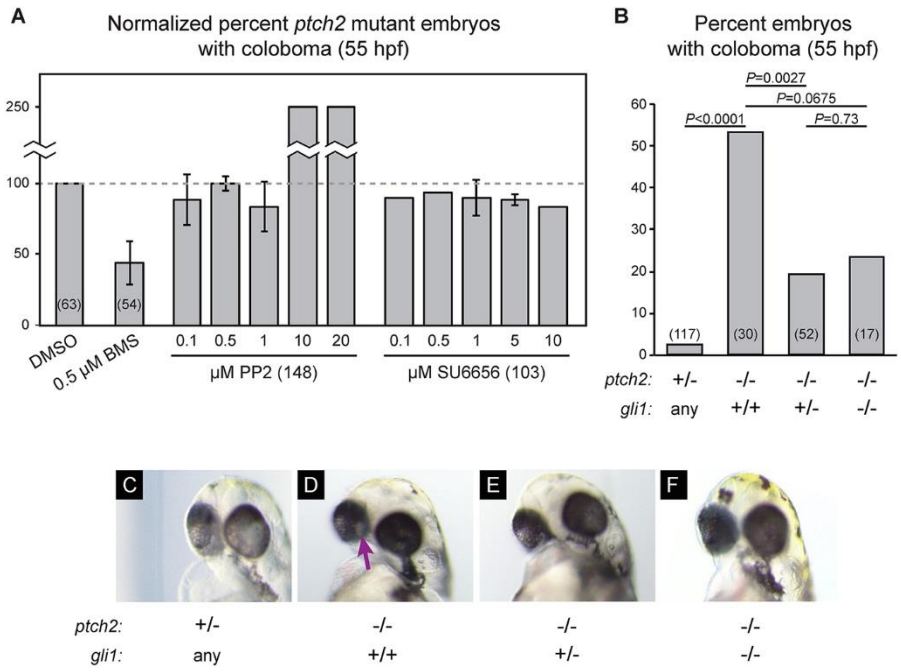


Figure 7. Canonical signaling via Gli1 is required for coloboma in the *ptch2*^{tc294z} mutant.

(A) Inhibition of non-canonical Hedgehog signaling via Src-family kinases (SFKs) does not rescue coloboma in the *ptch2*^{tc294z} mutant. Graph shows normalized percent of *ptch2*^{tc294z} mutant embryos.

(B) Genetic loss of one or both copies of *gli1* (using *gli1*^{ts269}) partially rescues coloboma in the *ptch2*^{tc294z} mutant. Quantification of percent embryos with coloboma. Numbers in parentheses indicate N.

(C-F) Representative images of embryos from the genetic experiments in (B), 55 hpf. (C) *ptch2*$^{+/-}$; *gli1*^{any}. (D) *ptch2*$^{-/-}$; *gli1*$^{+/+}$. (E) *ptch2*$^{-/-}$; *gli1*$^{+/-}$. (F) *ptch2*$^{-/-}$; *gli1*$^{-/-}$. Arrow, coloboma.

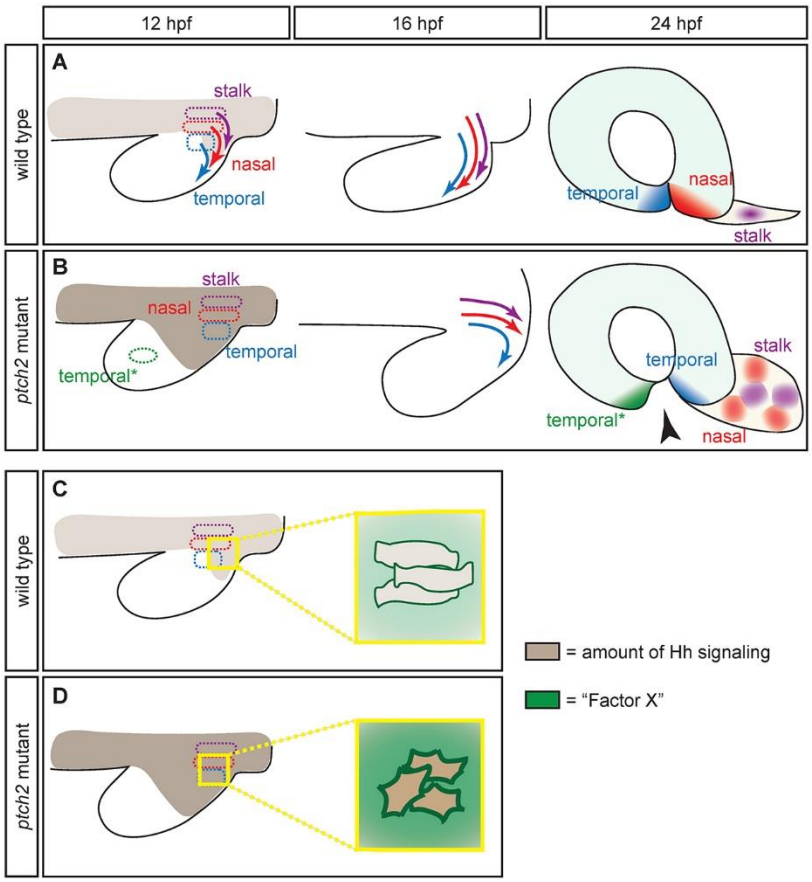


Figure 8. Model for optic fissure and stalk cell movements and regulation by Hh signaling.

(A) Optic fissure and stalk cell movements in wild type: adjacent cell populations (from origins marked as dashed circles) move along a J-shaped trajectory to occupy the optic fissure margins and stalk. (B) Optic fissure and stalk cell movements in the *ptch2*^{tc294z} mutant: cells that should normally contribute to the optic cup are altered in their movement and do not achieve their correct position.

(C) Proposed model for Factor X, a non-cell-autonomously acting cell surface or secreted molecule and downstream target of Hedgehog signaling. Under wild type conditions, Factor X is expressed at a level to appropriately regulate optic fissure and stalk cell morphology and orientation. (D) In the *ptch2*^{tc294z} mutant, overexpression and expansion of Factor X leads to diminished elongation and defective motility.

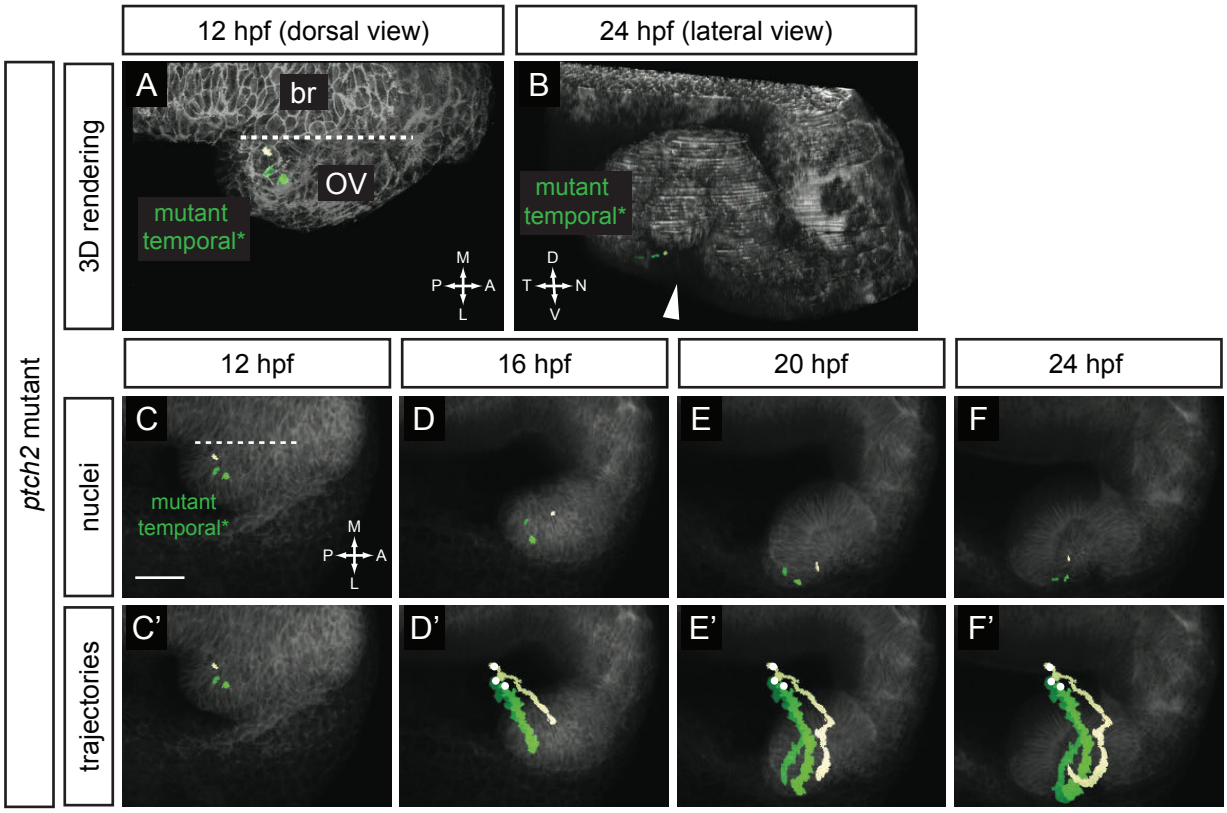


Figure S1. Origin and movement of cells contributing to the temporal optic fissure margin in the *ptch2*^{tc294z} mutant.
(A-F') *ptch2*^{tc294z} mutant temporal optic fissure cell movements (12-24 hpf). (A) Rendering of nuclei and membrane channel, 12 hpf, dorsal view. (B) Rendering of nuclei and membrane channel, 24 hpf, lateral view. (C-F) Nuclei over membrane channel average projection. (C'-F') Trajectories over membrane channel average projection. Cells execute a novel trajectory. Scale bar, 50 μm.

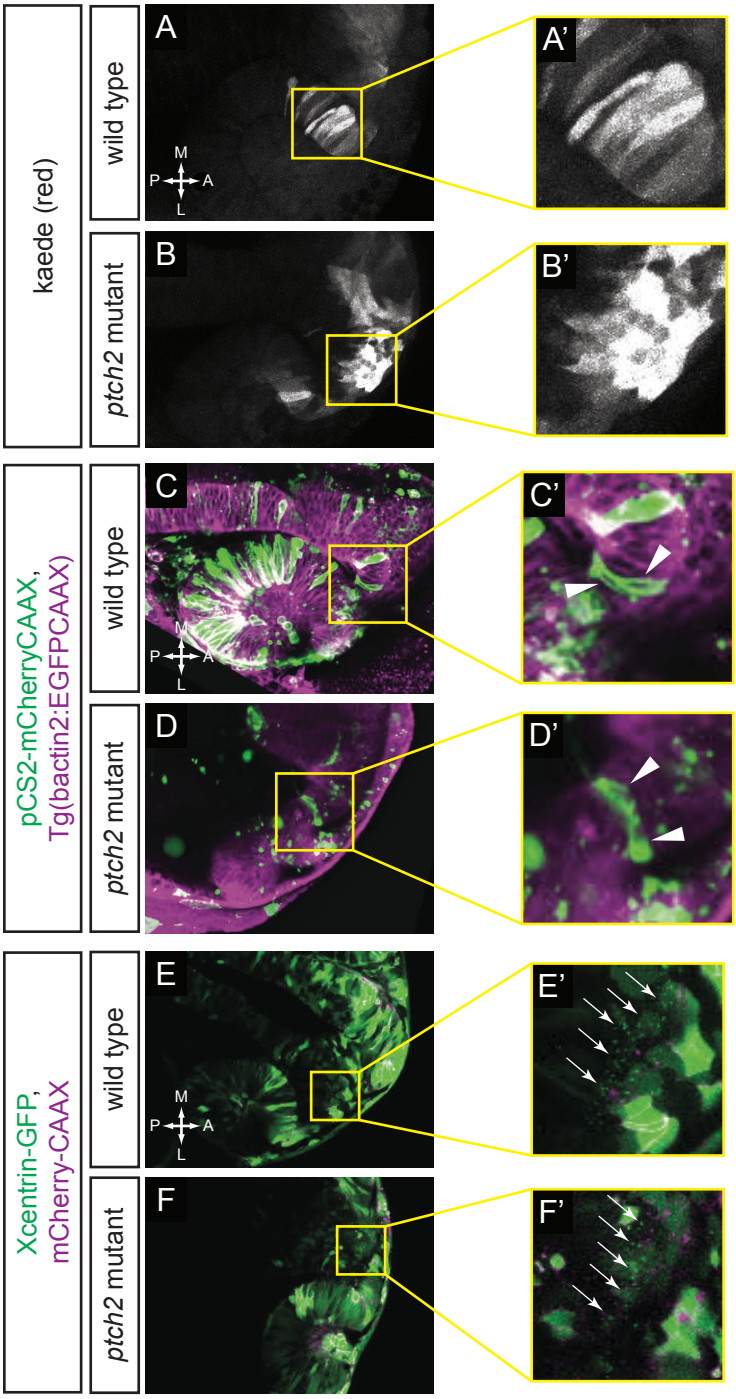


Figure S2. Single cell morphologies and cell polarity.
(A-B') Cell morphologies from embryos labeled by Kaede photoactivation (Figure 4). (A', B') Zoomed view of cell morphologies.
(C-D') Cell morphologies from embryos mosaically labeled via DNA injection (pCS2-mCherryCAAX; *green*) in a ubiquitously labeled background (Tg(bactin2:EGFPCAAX); *magenta*). Individual cells in wild type and *ptch2*^{tc294z} mutant optic stalks were analyzed for cell morphology (roundness and elongation angle). (C',D') Zoomed view of cell morphologies.
(E-F') Centrosomes (EGFP-Xcentrin; *green*) in wild type (E) and *ptch2*^{tc294z} mutant (F) embryos. Membranes are labeled with mCherry-CAAX (*magenta*; brightness is decreased so that centrosomes are visible). Centrosomes are oriented toward the apical surface of the optic stalk in both wild type and mutant embryos. (E',F') Zoomed view of oriented centrosomes (arrows). Note that EGFP-Xcentrin labels centrosomes, but overexpression due to RNA injection yields additional mosaic cytoplasmic fluorescence.

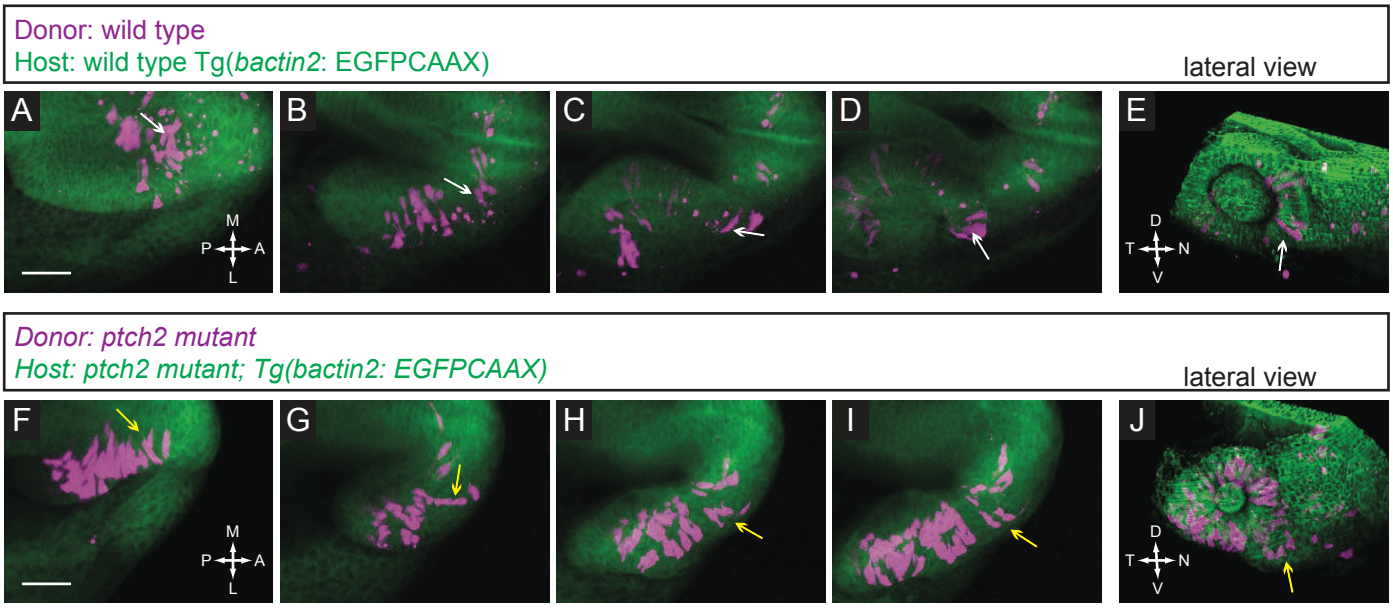
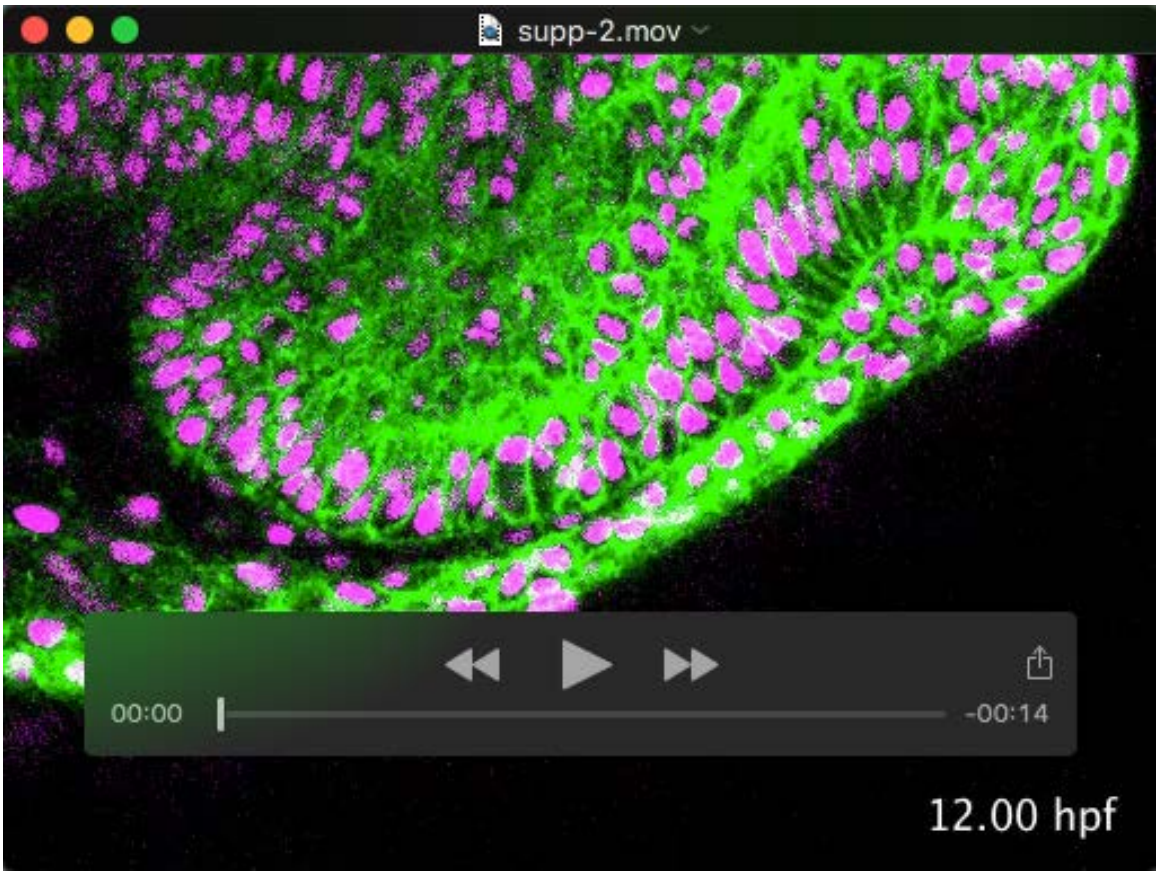


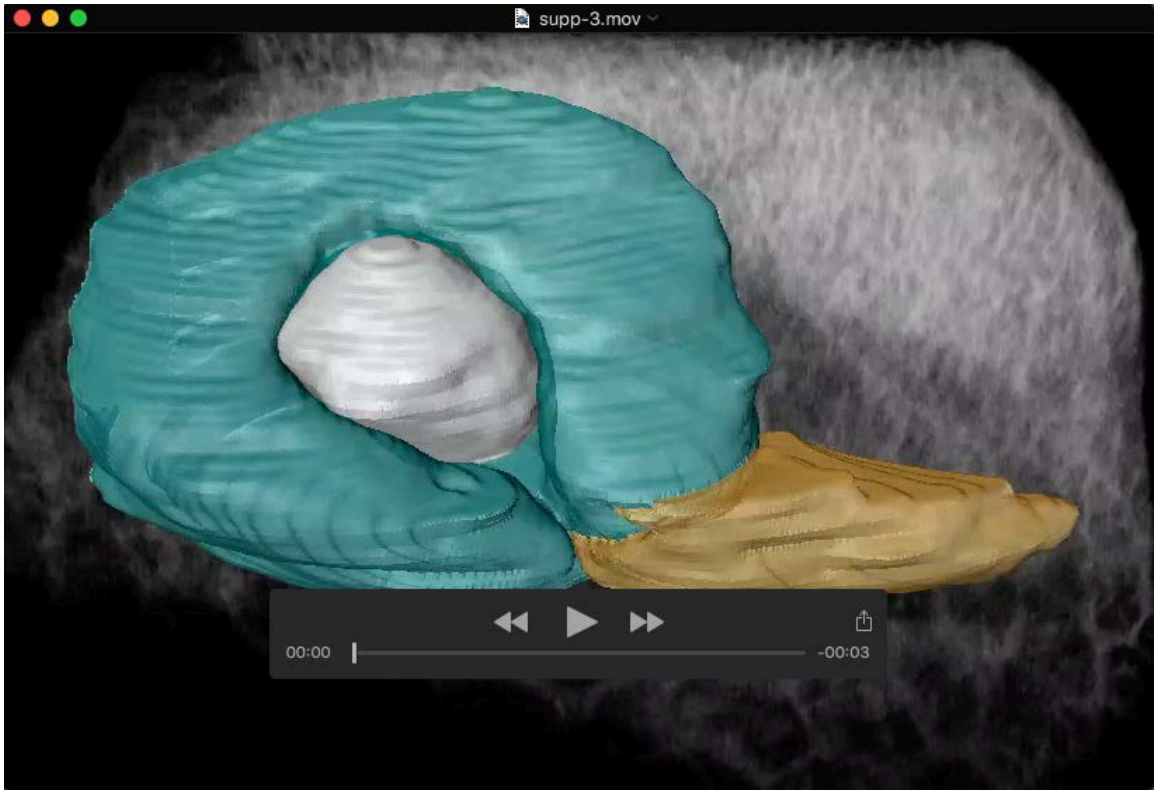
Figure S3. Localization and movement of transplanted cells in control conditions.
(A-E) Transplantation of wild type donor cells into a wild type host. (A-D) Donor cells move out of the midline region, through the prospective optic stalk, and into the optic cup. Note the bipolar, elongated morphology of the cell contributing to the nasal optic fissure (*white arrows*). (E) Lateral view of 3D rendering of final timepoint.
(F-J) Transplantation of *ptch2*^{tc294z} mutant donor cells into a *ptch2*^{tc294z} mutant host. (F-I) Many donor cells in this movie start already within the optic vesicle. Following cells that emerge from the midline region, some cells exhibit and maintain this morphology through the optic stalk and into the optic cup. Other cells (*yellow arrows*) lose their bipolar morphology and contribute to the optic stalk. (J) Lateral view of 3D rendering of final timepoint.
Scale bar, 50 μm.

Table S1. Summary of Transplant Data

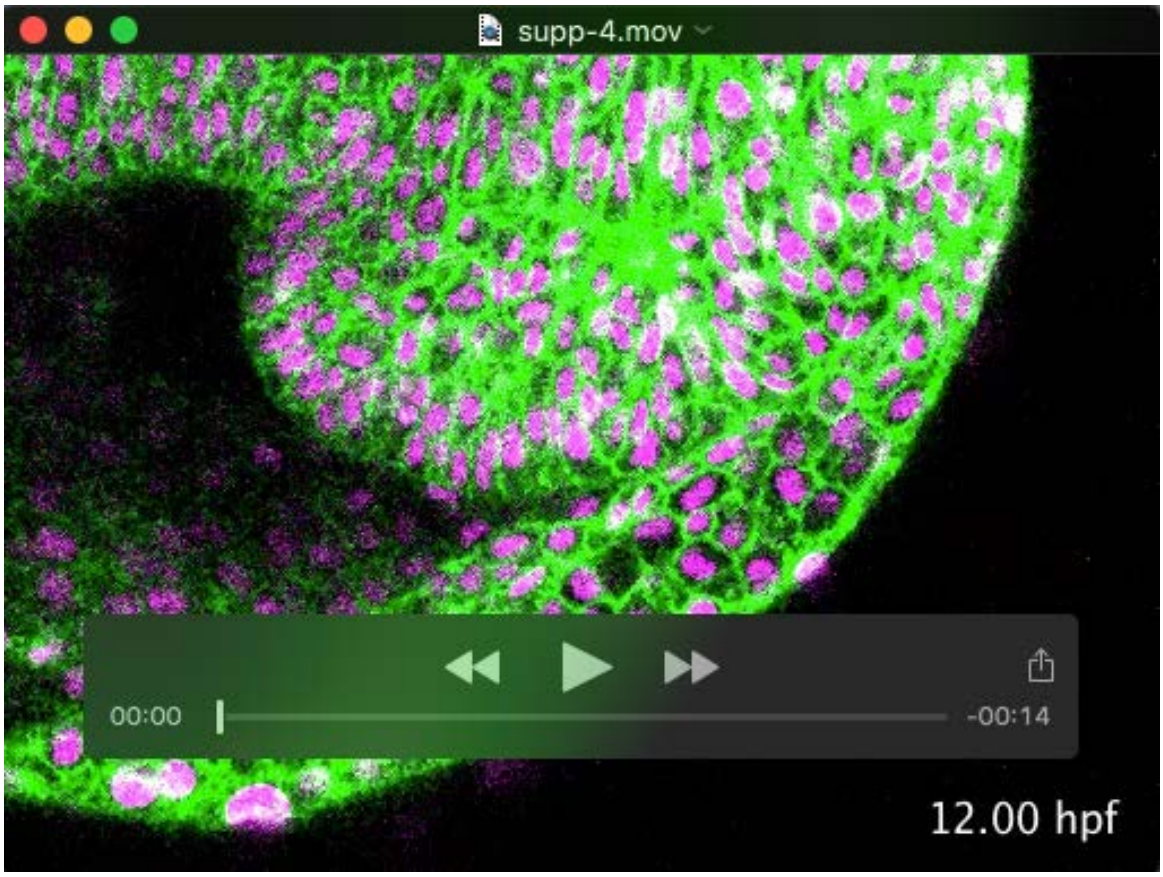
			MIDLINE STARTING POSITION	ENDING POSITION			
	experiment	D>H genotypes	total # cells in ROI	Nasal Optic Fissure	Cell Morphology	Optic Stalk	Cell Morphology
wt>wt	160601pos1	wt>wt	1	1	elongated	0	
	151104pos1	wt>het	7	1 group	most are elongated	1	most are elongated
	160601pos2	wt>wt	2	0		2	elongated
	160916pos2	wt>wt	3	3	2 elongated, 1 aberrant	0	
		SUBTOTALS	13	9		3	
mut>wt	160115pos1	mut>wt	2	0		2	elongated
	160115pos3	mut>wt	4	4	elongated	0	
	160212pos3	mut>wt	1	1	elongated	0	
	160308pos4	mut>wt	1	1	elongated	0	
	160308pos3	mut>wt	4	0		4	elongated
	160512pos3	mut>het	2	1	elongated	1	elongated
	160525pos2	mut>het	9	6	elongated	3	elongated
		SUBTOTALS	23	13		10	
wt>mut	170127pos2	wt>mut	6	2		4	
	170127pos3	wt>mut	12	5		7	
	170127pos4	wt>mut	8	0		8	
	170127pos5	wt>mut	5	1		4	
	160512pos4	wt>mut	7	0		3 indiv, 4 as a clump	aberrant
	161004pos3	wt>mut	2	1	larger cell, elongated	1	smaller cell, aberrant
	170126pos1	wt>mut	1	0		1	aberrant
		SUBTOTALS	41	9		32	
mut>mut	160421pos1	mut>mut	4	2		0	
	170209pos2	mut>mut	11	5	elongated	clump of 2 near a clump of 4	aberrant
	170216pos1	mut>mut	2	0		2	aberrant
	170216pos3	mut>mut	2	0		2	aberrant
	170216pos4	mut>mut	4	0		4	1 elongated, 3 aberrant
		SUBTOTALS	23	7		14	



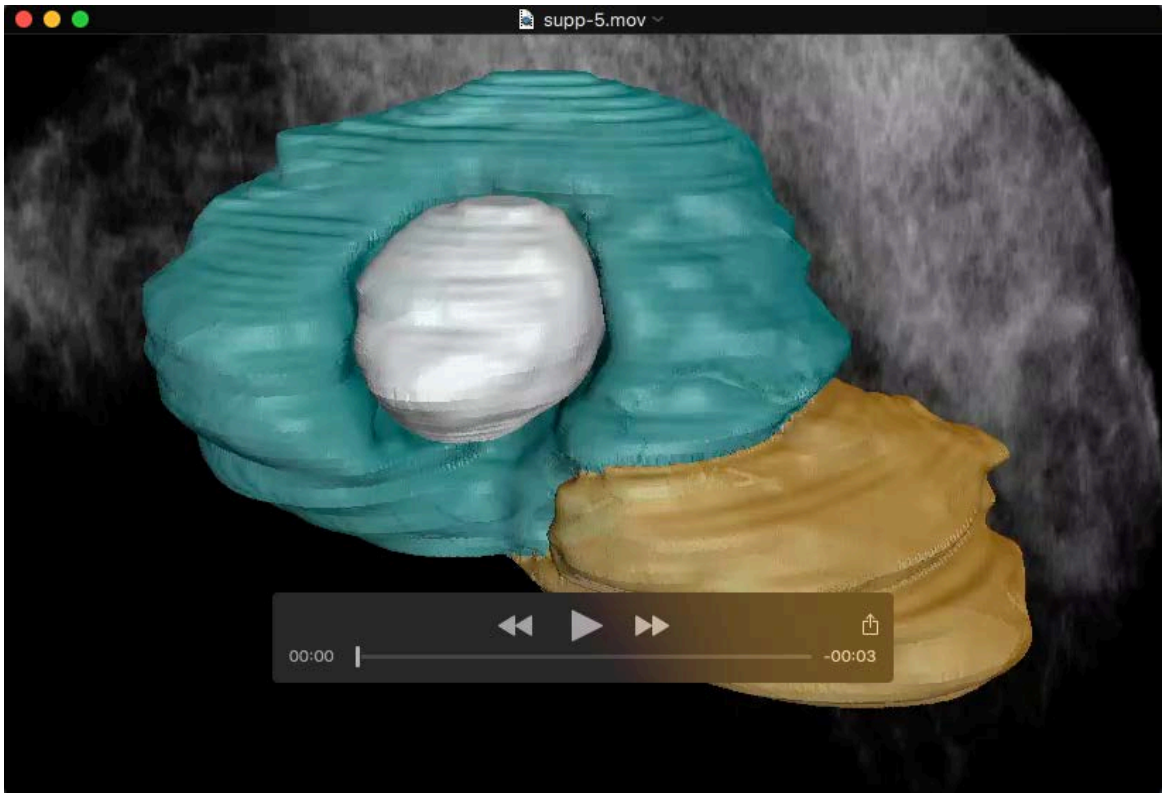
Movie 1. Timelapse of wild type optic cup morphogenesis. Single confocal section from a 4D data set showing right optic vesicle, ~12-24 hpf. Time interval between z-stacks, 2.75 minutes. Dorsal view. Embryo labeled with EGFP-CAAX (membranes, *green*) and H2A.F/Z-mCherry (nuclei, *magenta*).



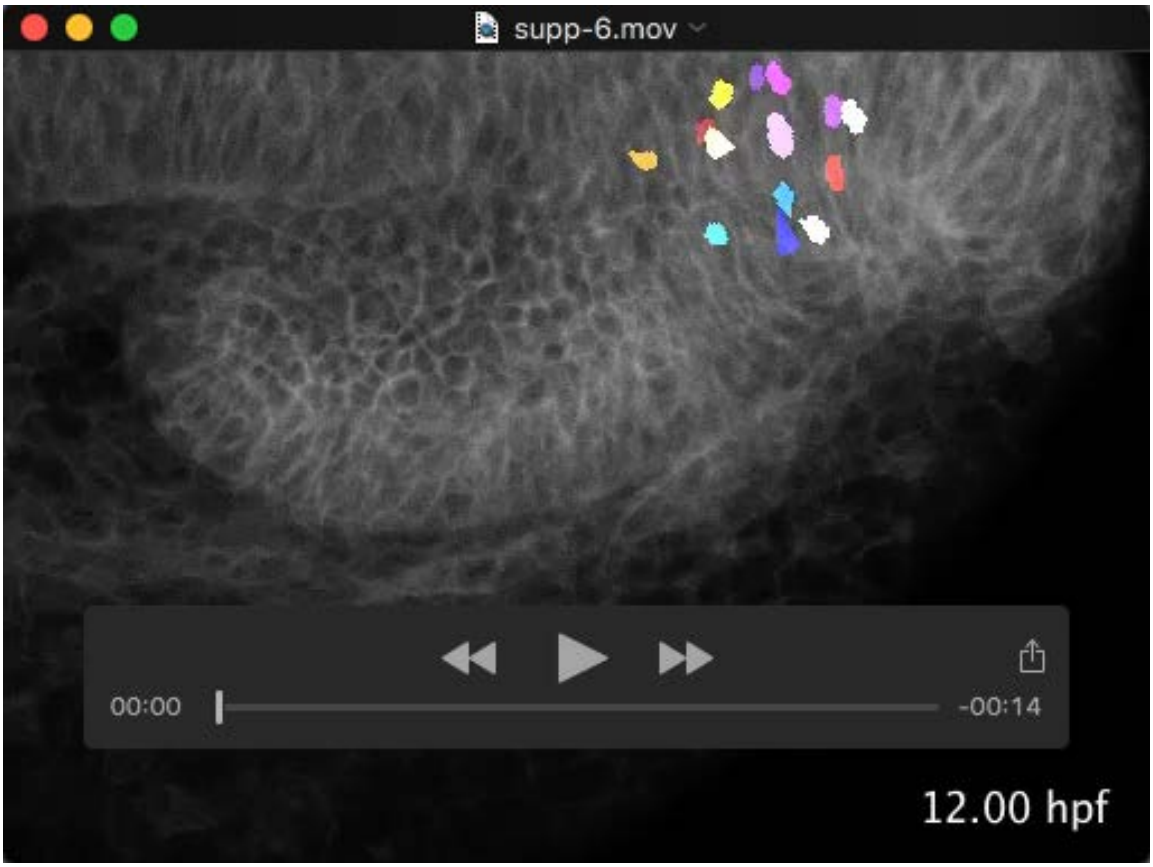
Movie 2. Segmentation of wild type optic cup to visualize normal optic fissure morphology. Lateral view of 3-dimensional volume rendering of manually segmented optic cup (*teal*), lens (*gray*), and optic stalk (*gold*), 24 hpf. The optic fissure margins are closely apposed at the ventral side of the optic cup.



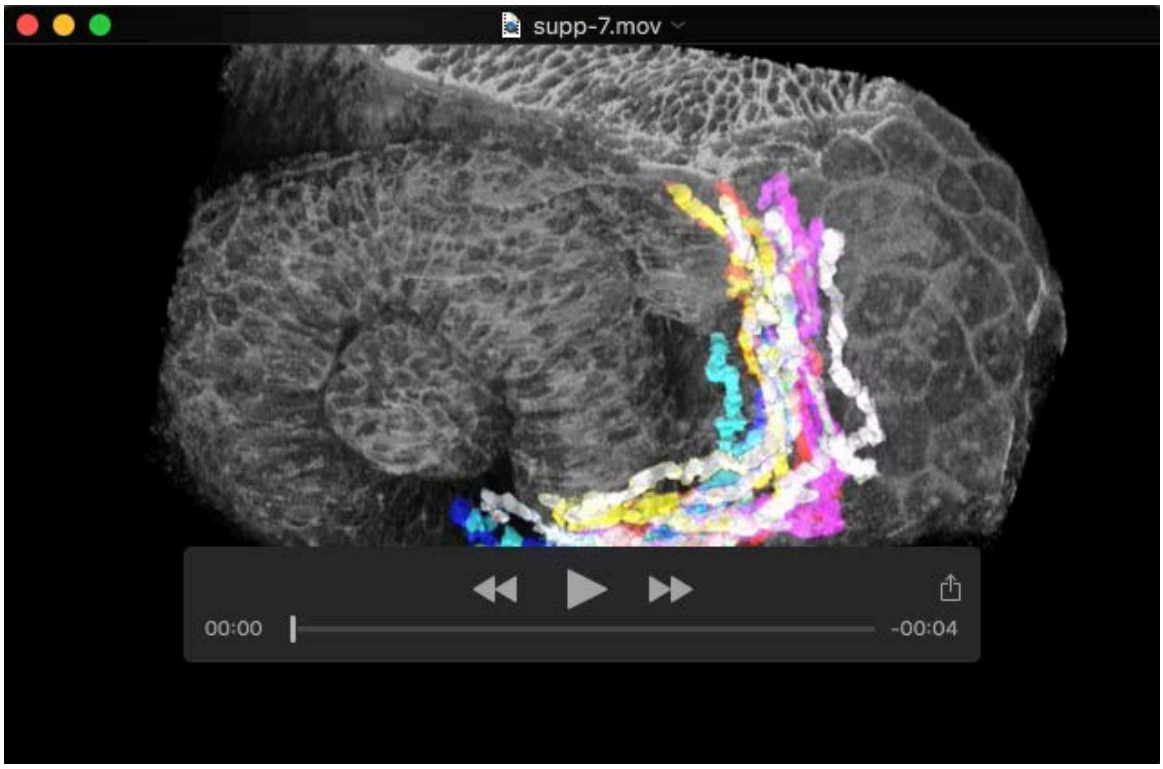
Movie 3. Timelapse of *ptch2*^{tc294z} mutant optic cup morphogenesis. Single confocal section from a 4D data set showing right optic vesicle, ~12-24 hpf. Time interval between z-stacks, 2.75 minutes. Dorsal view. Embryo labeled with EGFP-CAAX (membranes, *green*) and H2A.F/Z-mCherry (nuclei, *magenta*).



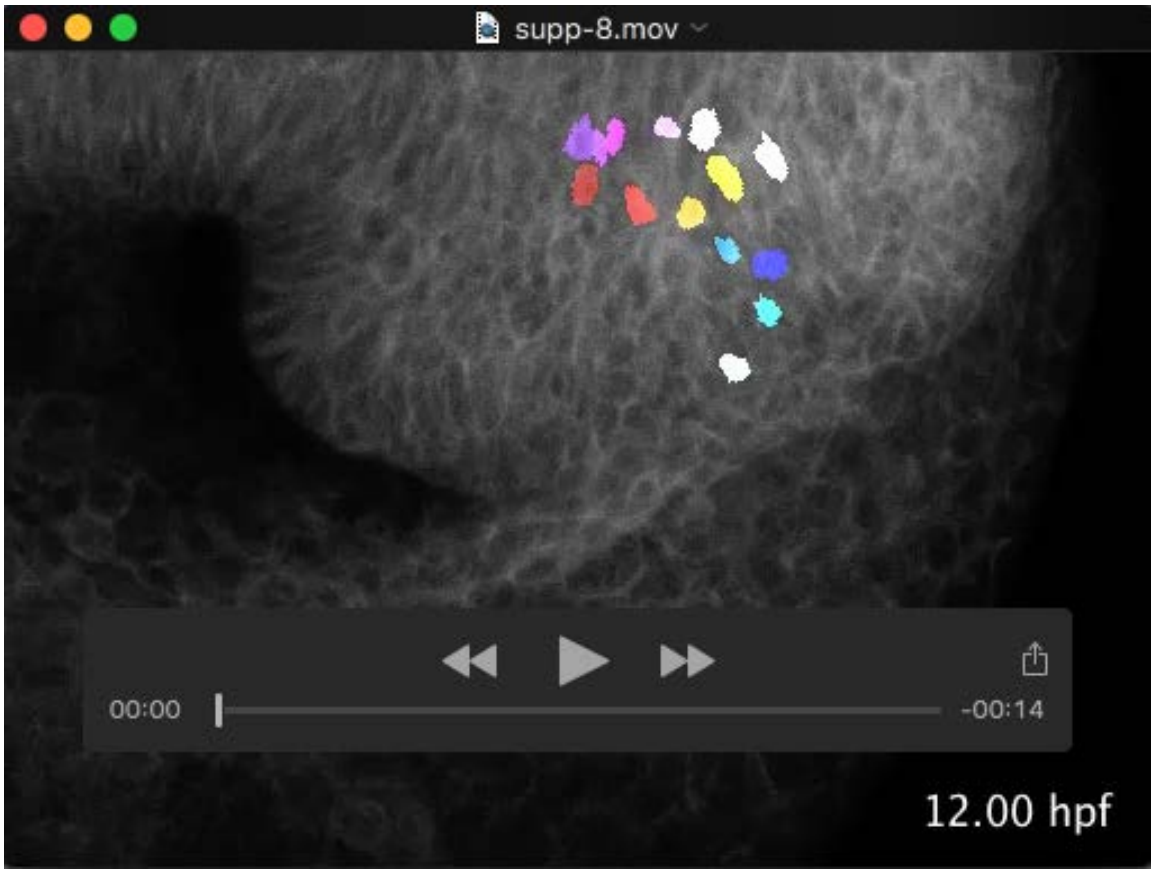
Movie 4. Segmentation of *ptch2*^{tc294z} mutant optic cup to visualize disrupted optic fissure morphology. Lateral view of 3-dimensional volume rendering of manually segmented optic cup (*teal*), lens (*gray*), and optic stalk (*gold*), 24 hpf. The optic fissure has not formed correctly; there are no obvious, closely apposed margins in the ventral optic cup.



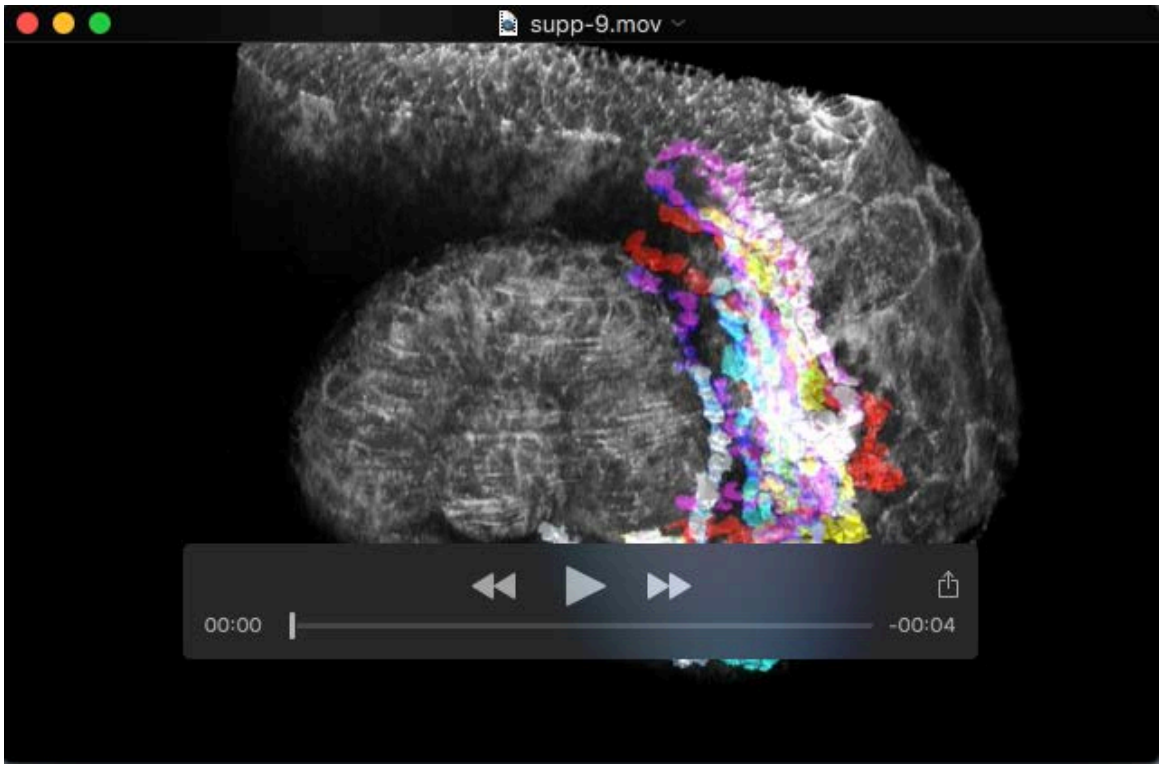
Movie 5. Wild type trajectories (~12-24 hpf): cells moving to optic fissure margins and optic stalk. Trajectories superimposed on average projection of membrane channel (*grayscale*), with temporal nuclei (*blue shades*), nasal nuclei (*red/yellow shades*), and stalk nuclei (*purple shades*). Time interval between z-stacks, 2.75 minutes. Dorsal view.



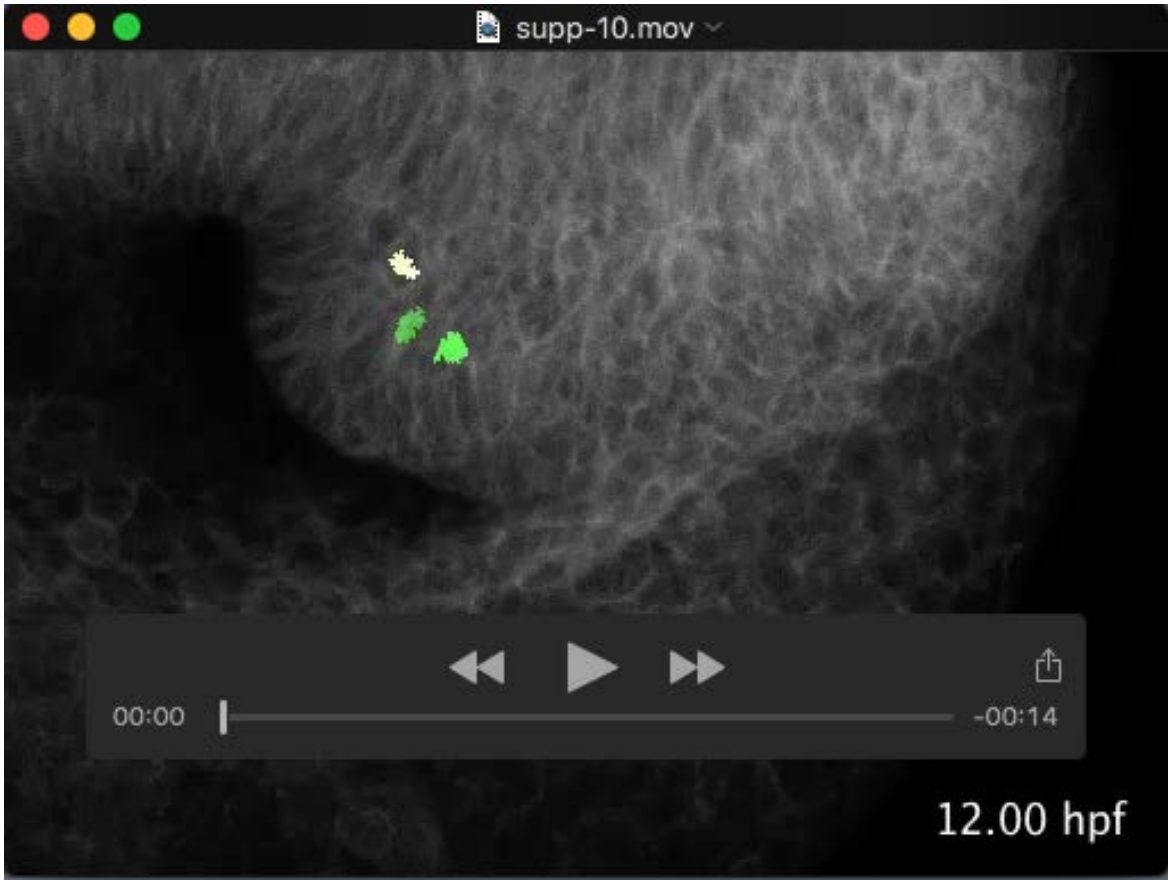
Movie 6. Rotation of wild type optic fissure and stalk trajectories. Trajectories display 3-dimensional movement.



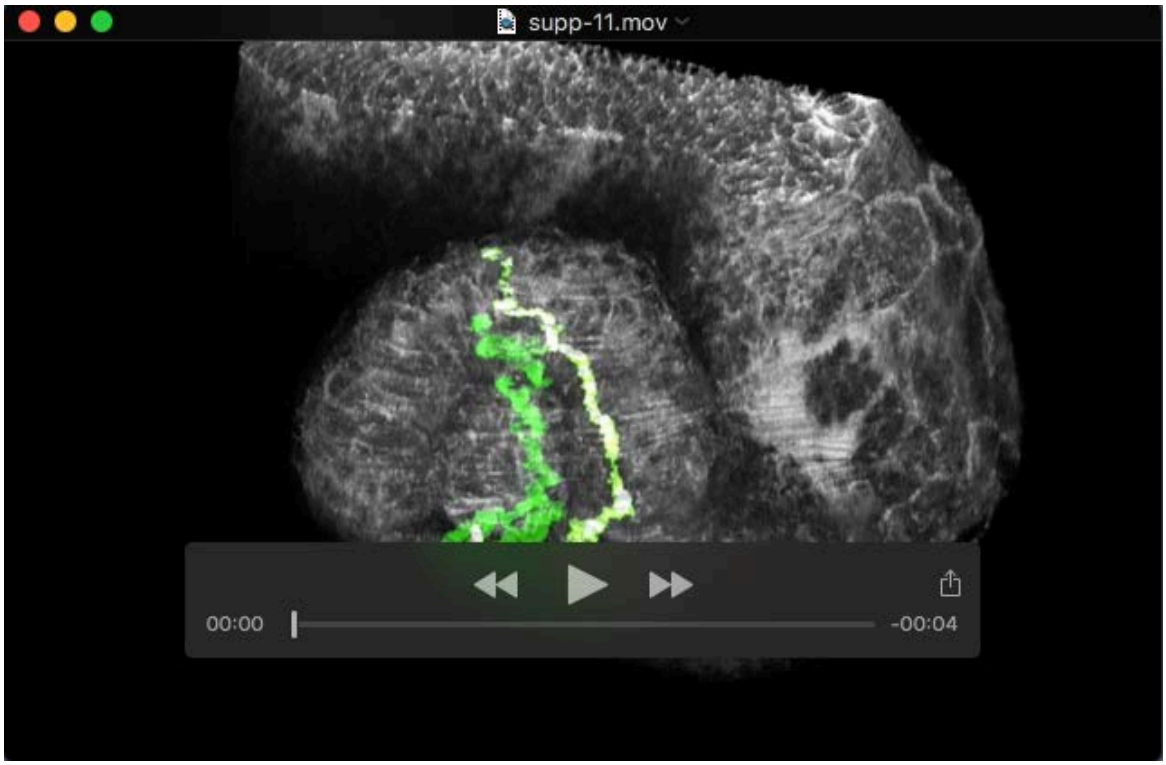
Movie 7. *ptch2*^{tc294z} mutant trajectories (~12-24 hpf): cells moving to disrupted optic fissure and optic stalk. Trajectories superimposed on average projection of membrane channel (*grayscale*), with temporal nuclei (*blue shades*), nasal nuclei (*red/yellow shades*), and stalk nuclei (*purple shades*). Time interval between z-stacks, 2.75 minutes. Dorsal view.



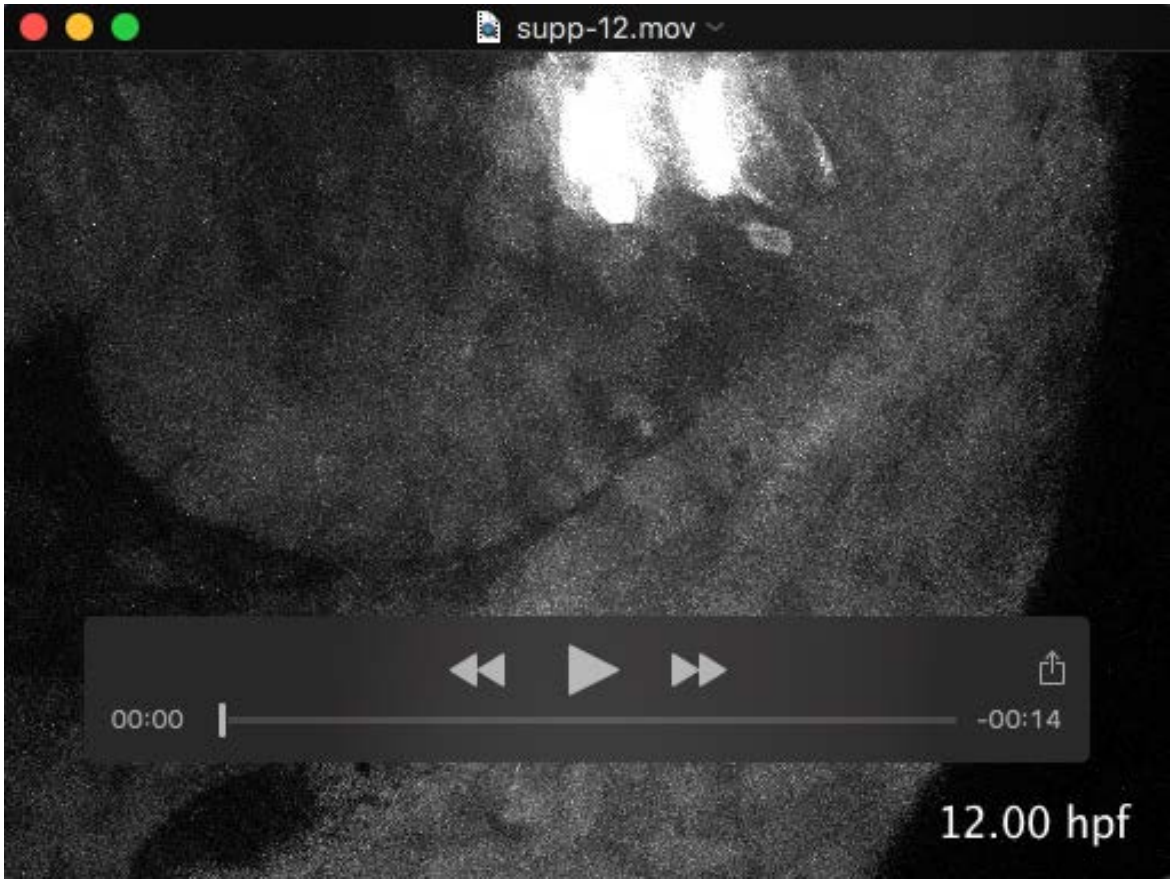
Movie 8. Rotation of *ptch2*^{tc294z} mutant trajectories from prospective optic fissure and stalk origin. Trajectories display 3-dimensional movement.



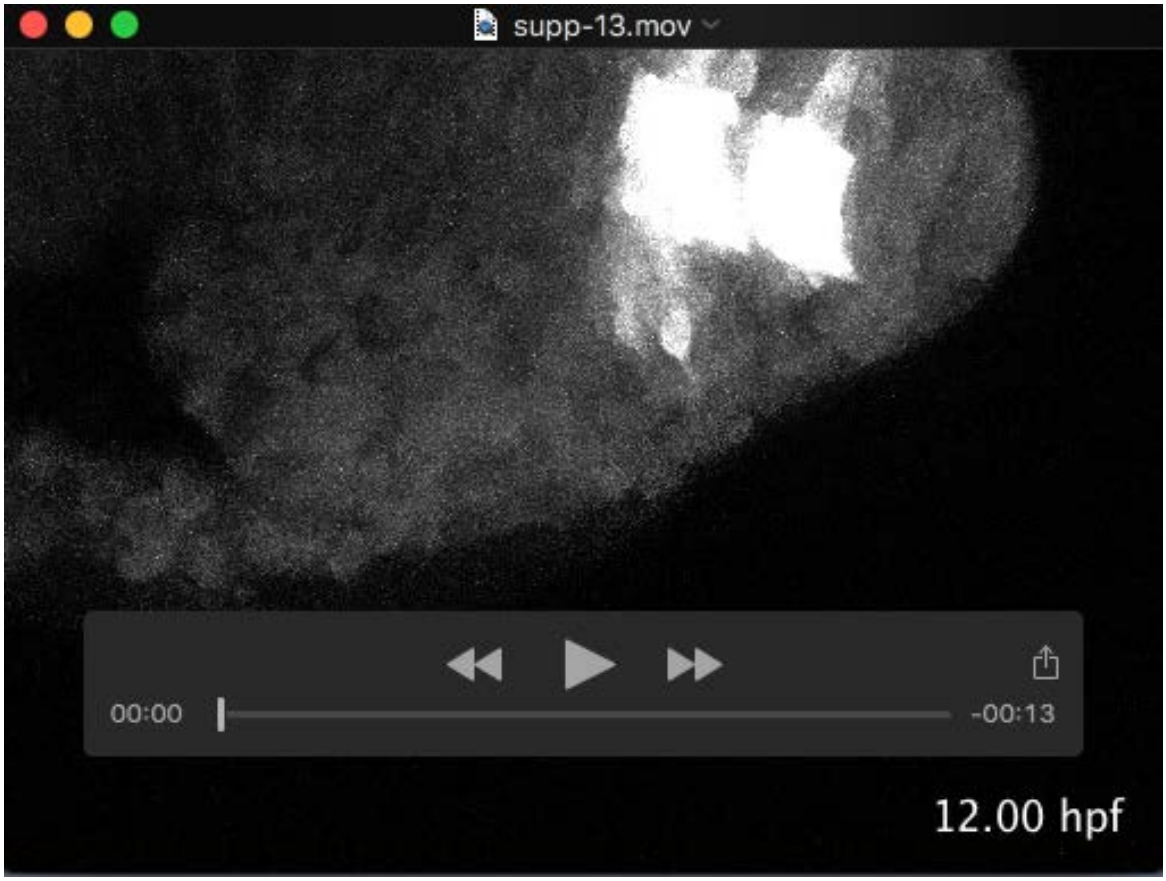
Movie 9. *ptch2*^{tc294z} mutant trajectories (~12-24 hpf): cells moving to the aberrant temporal margin. Trajectories superimposed on average projection of membrane channel (*grayscale*), with actual mutant temporal nuclei (*green shades*). Time interval between z-stacks, 2.75 minutes. Dorsal view.



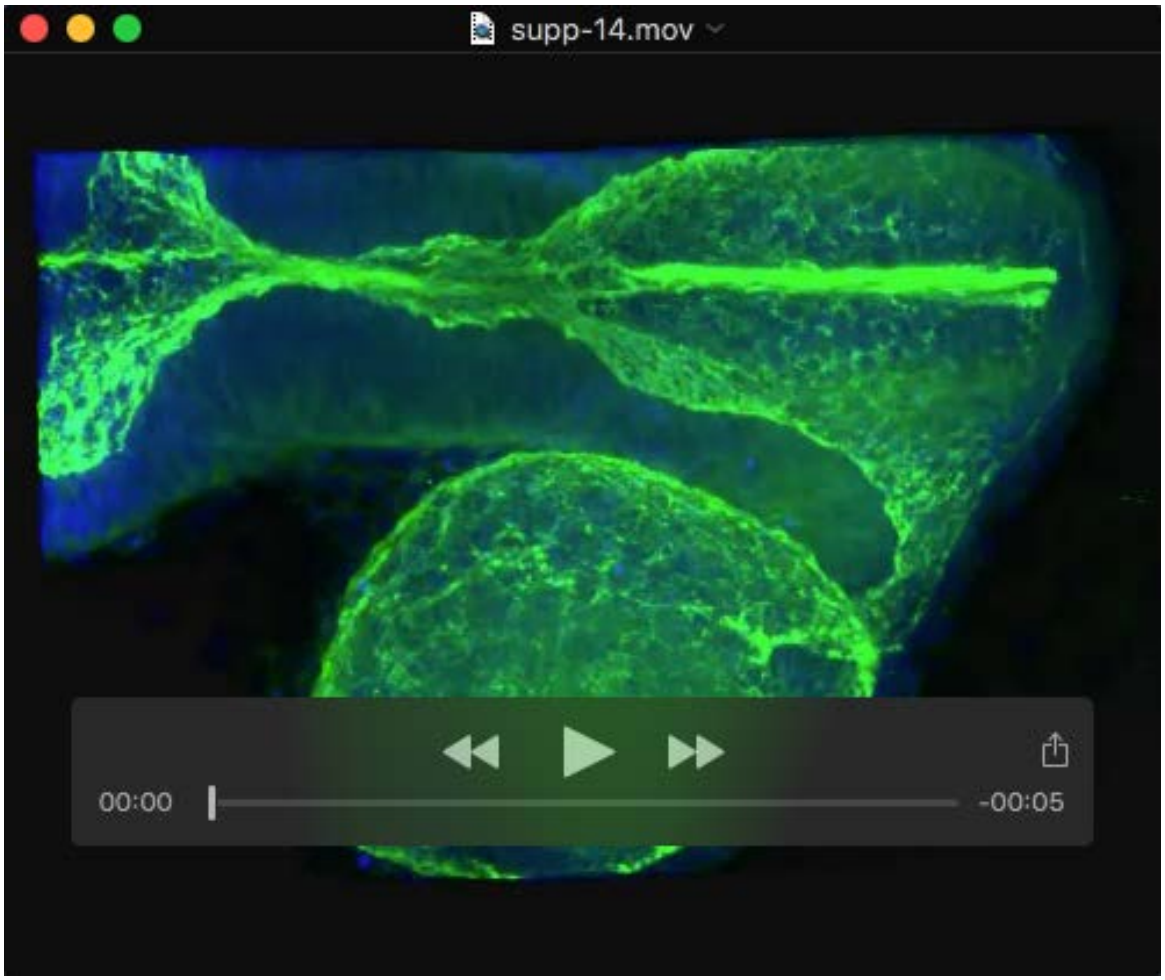
Movie 10. Rotation of *ptch2*^{tc294z} mutant trajectories of cells contributing to disrupted temporal margin. Trajectories display 3-dimensional movement.



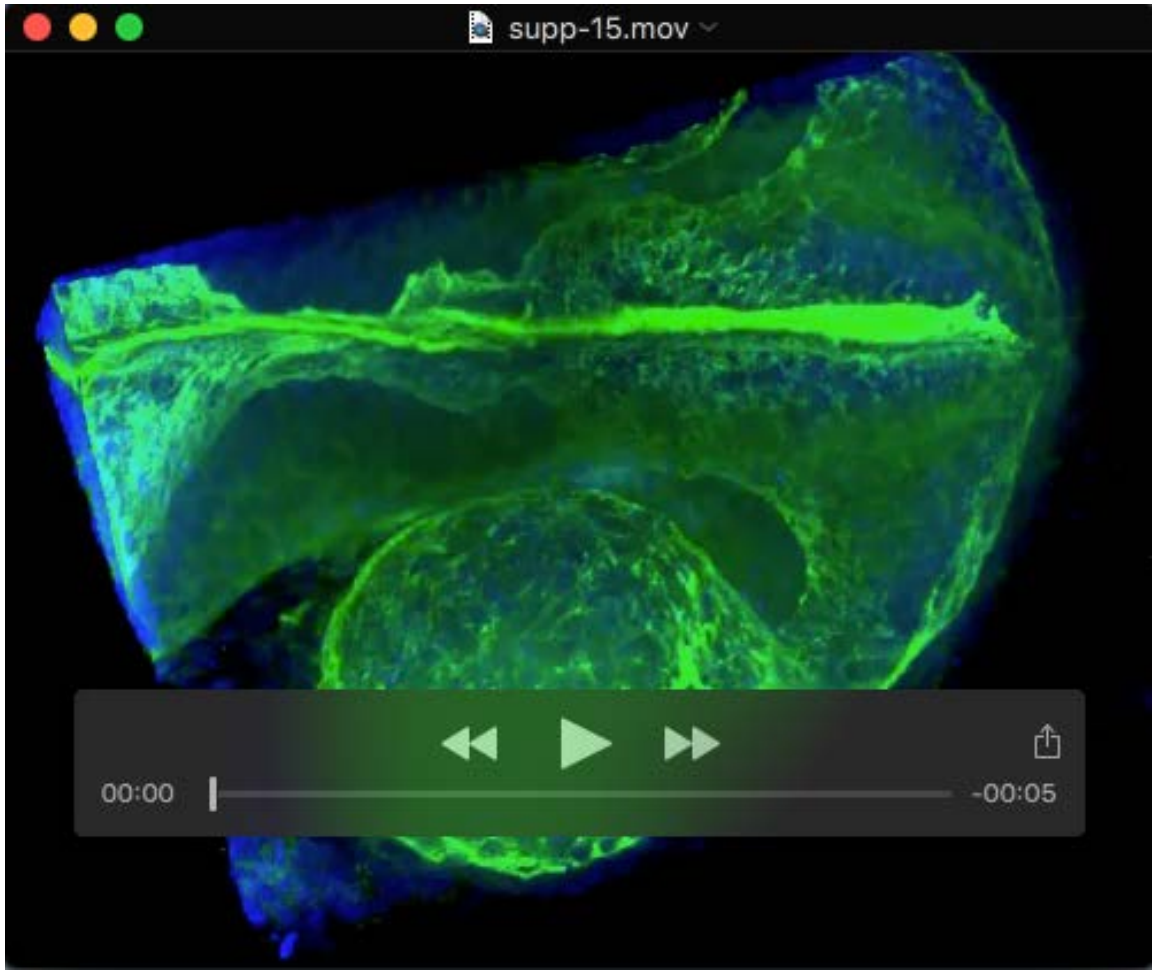
Movie 11. Timelapse of prospective nasal optic fissure cell behaviors during wild type optic cup morphogenesis (12-24 hpf). Marked cells exit the midline region, move through the prospective optic stalk domain, and into the ventronasal optic cup and nasal margin of the optic fissure. Cells maintain an elongated morphology throughout the process. Maximum intensity projection of photoconverted Kaede (grayscale) to visualize entire cell morphologies. Time interval between z-stacks, 2.75 minutes. Dorsal view.



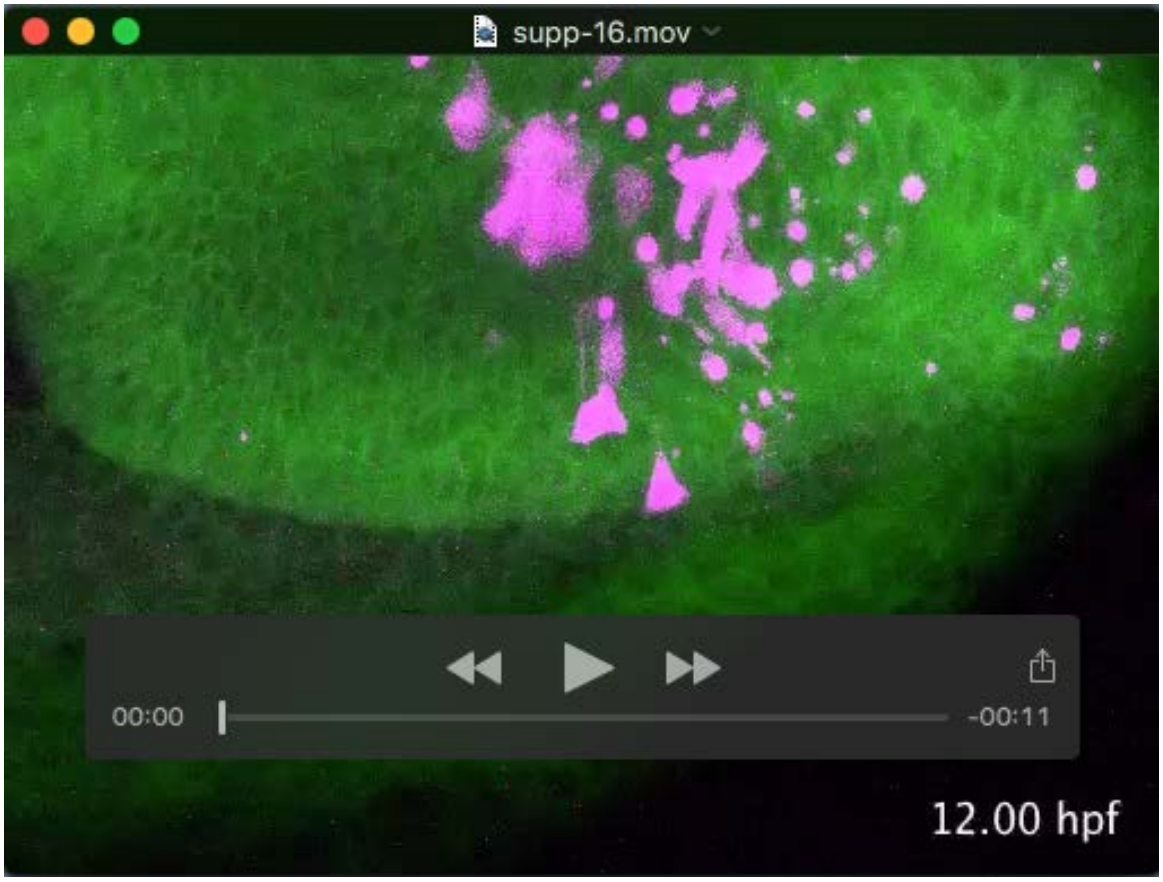
Movie 12. Timelapse of cell movements in the *ptch2*^{tc294z} mutant (12-24 hpf). Marked cells originate from a region that should contribute to the ventronasal optic cup and nasal margin of the optic fissure. Cells exit the midline region, initially exhibit an elongated morphology, but then take on an aberrant morphology and contribute to the optic stalk region. Maximum intensity projection of photoconverted Kaede (*grayscale*) to visualize entire cell morphologies. Time interval between z-stacks, 2.75 minutes. Dorsal view.



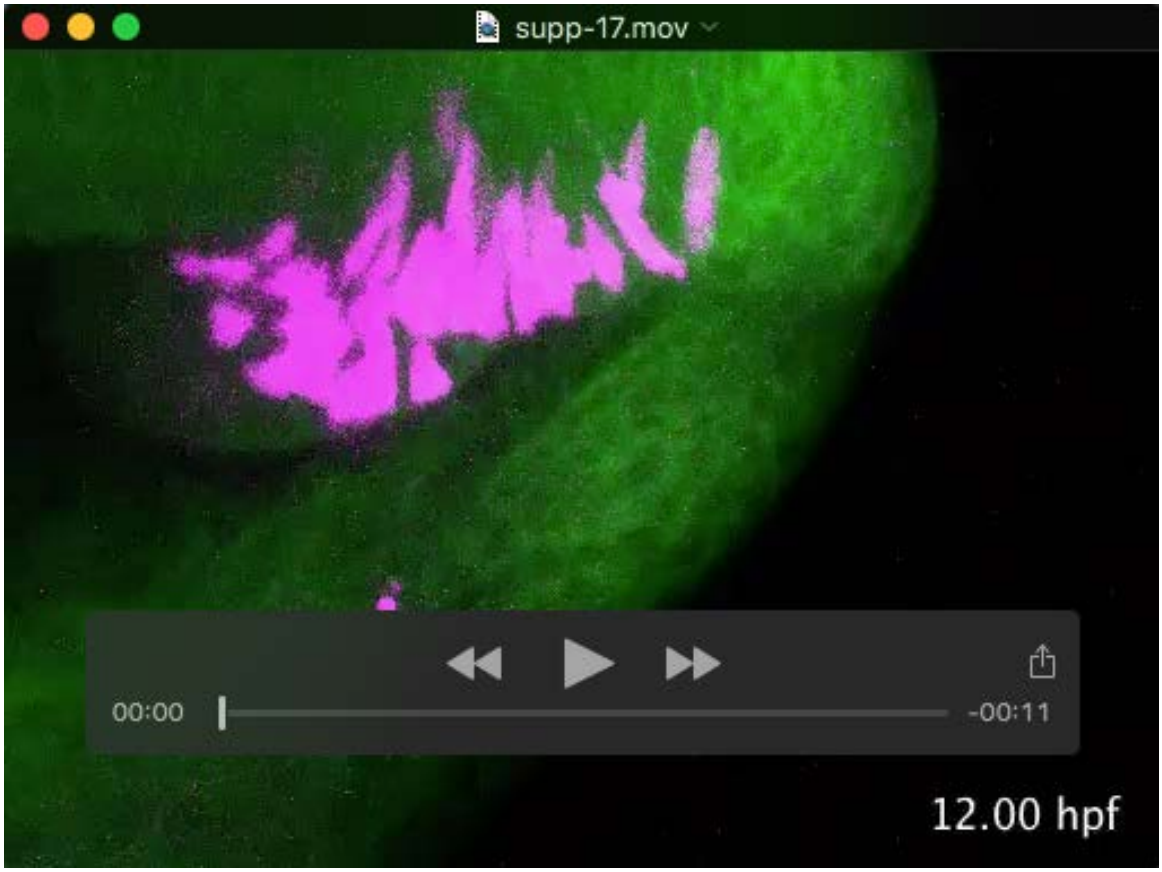
Movie 13. Rotation of wild type embryo stained for aPKC (24 hpf). A coherent apical surface is apparent in the optic cup, optic stalk, and prospective brain.



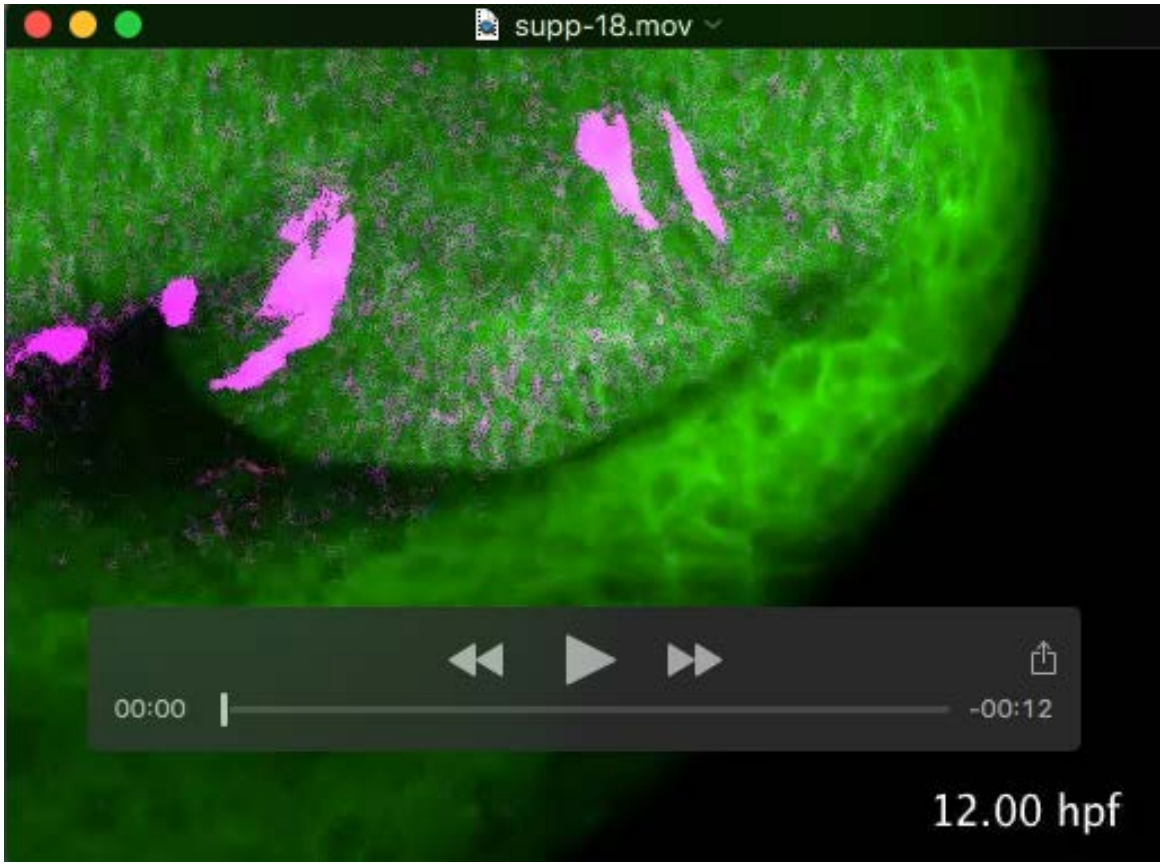
Movie 14. Rotation of *ptch2*^{tc294z} mutant embryo stained for aPKC (24 hpf). A coherent apical surface is still apparent in the optic cup, larger optic stalk, and prospective brain.



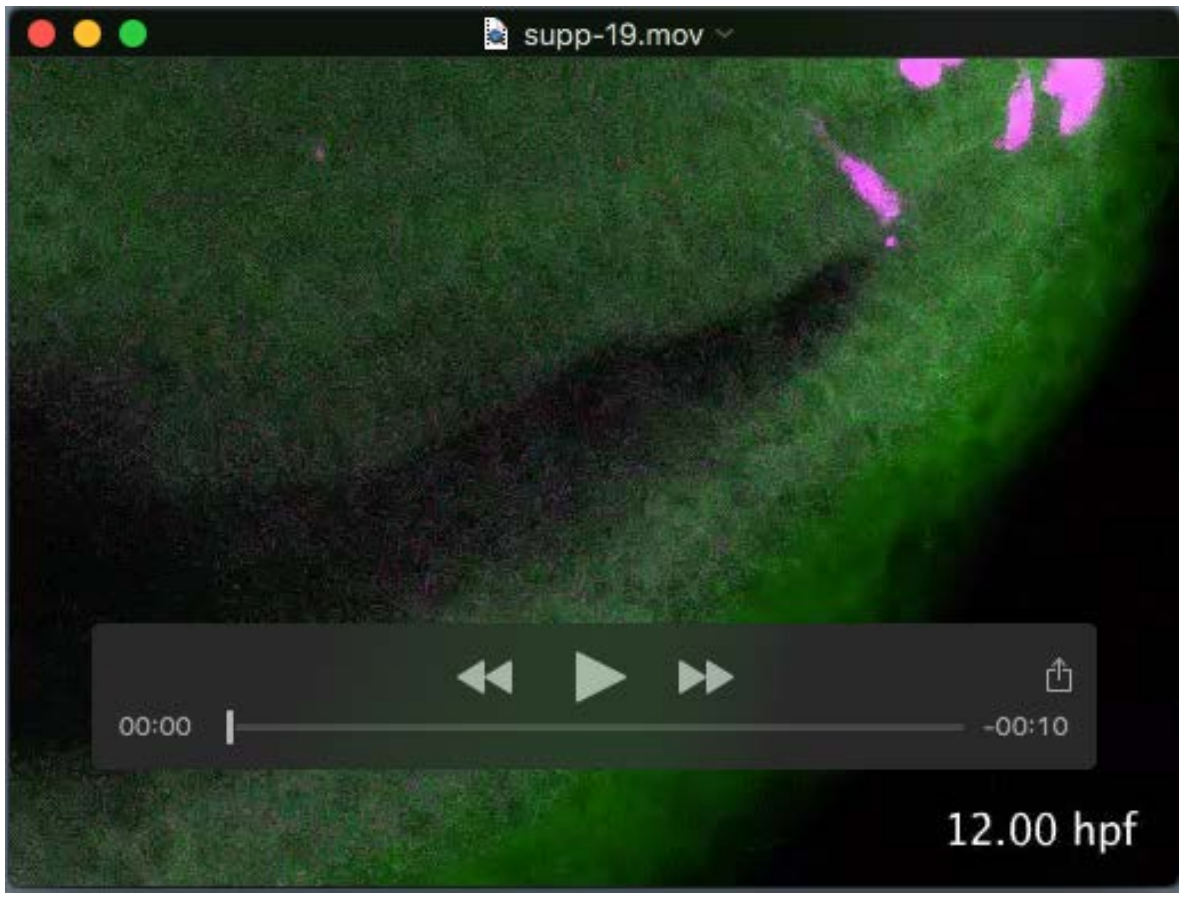
Movie 15. Timelapse of wild type donor cells in a wild type host. Donor cells (*magenta*) originating in the midline region move through the prospective optic stalk and into the optic cup. Cells maintain a bipolar morphology, and one cell contributes to the nasal optic fissure margin. Maximum intensity projection of donor cells (*magenta*, fluorescent dextran lineage tracer) superimposed on average projection of host membrane label (*green*, Tg(*bactin2*:EGFPCAAX)). Time interval between stacks, 6.25 minutes. Dorsal view.



Movie 16. Timelapse of *ptch2*^{tc294z} mutant donor cells in a *ptch2*^{tc294z} mutant host. At the start of the movie, donor cells (*magenta*) are found within the optic vesicle and the midline region; we specifically follow those originating from the midline region. These cells exit the midline, initially exhibiting a bipolar morphology, however, they lose their bipolar morphology and exhibit a variety of morphologies within the optic stalk, none of which are observed in the wild type optic stalk. Maximum intensity projection of donor cells (*magenta*, fluorescent dextran lineage tracer) superimposed on average projection of host membrane label (*green*, Tg(*bactin2*:EGFPCAAX)). Time interval between stacks, 9.22 minutes. Dorsal view.

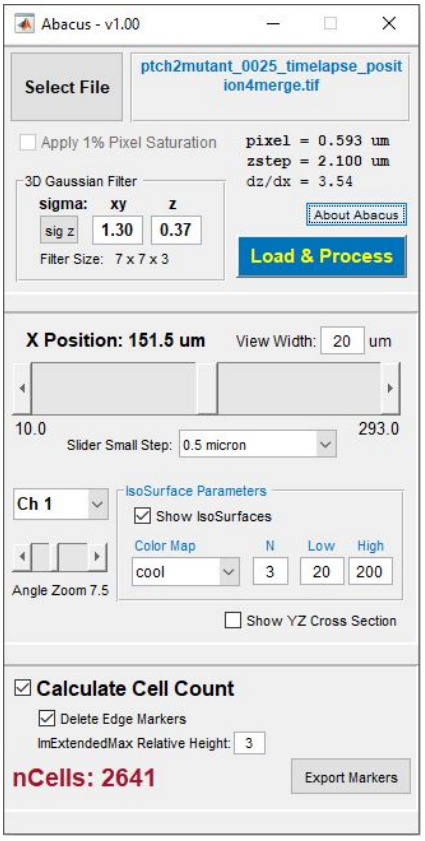


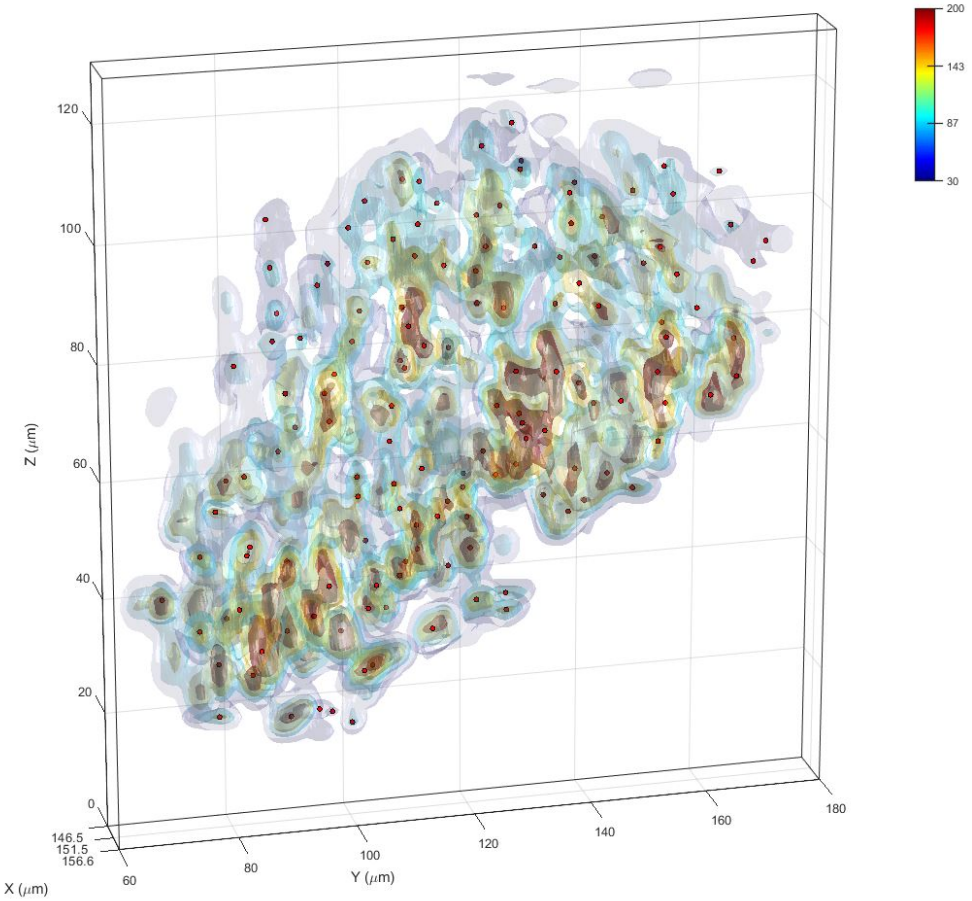
Movie 17. Timelapse of *ptch2*^{tc294z} mutant donor cells in a wild type host. Donor cells (*magenta*) originating in the midline region move through the prospective optic stalk and into the optic cup. Cells exhibit a bipolar morphology and one cell contributes to the optic fissure margin. Maximum intensity projection of donor cells (*magenta*, fluorescent dextran lineage tracer) superimposed on average projection of host membrane label (*green*, Tg(*bactin2*:EGFPCAAX)). Time interval between stacks, 8.8 minutes. Dorsal view.



Movie 18. Timelapse of wild type donor cells in a *ptch2*^{tc294z} mutant host. Donor cells (*magenta*) originating in the midline region contribute in a large group to the central retina, or lose bipolar morphology and contribute to the optic stalk. Maximum intensity projection of donor cells (*magenta*, fluorescent dextran lineage tracer) superimposed on average projection of host membrane label (*green*, Tg(*bactin2*:EGFPCAAX)). Time interval between stacks, 7.35 minutes. Dorsal view.

Supplementary File 1: Abacus custom MATLAB code for 3D cell counting





<<<< Abacus MATLAB code below >>>>

```
function varargout = abacus(varargin)
% ABACUS MATLAB code for abacus.fig
%      ABACUS, by itself, creates a new ABACUS or raises the existing
%      singleton*.
%
%      H = ABACUS returns the handle to a new ABACUS or the handle to
%      the existing singleton*.
%
%      ABACUS('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in ABACUS.M with the given input arguments.
%
%      ABACUS('Property','Value',...) creates a new ABACUS or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before abacus_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to abacus_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help abacus

% Last Modified by GUIDE v2.5 03-Aug-2018 11:39:42

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @abacus_OpeningFcn, ...
                   'gui_OutputFcn',  @abacus_OutputFcn, ...
                   'gui_LayoutFcn',  @abacus_LayoutFcn, ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before abacus is made visible.
function abacus_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

handles.figure1.Name = 'Abacus - v1.00';

handles.Apply_Saturation_CheckBox.Value = false;
handles.Apply_Saturation_CheckBox.Enable = 'off';
handles.Voxel_Dimension_Display.String = [{'pixel = 0.000 um'}; {'zstep = 0.000 um'}];
handles.Voxel_Dimension_Display.Enable = 'off';

set(handles.XYSigmaGaussFilt_Input,'Enable','off','String','0.00')
set(handles.ZSigmaGaussFilt_Input, 'Enable','off','String','0.00')

handles.text4.Enable = 'off';
handles.text5.Enable = 'off';
handles.text6.Enable = 'off';
handles.LoadAndProcessData_Button.Enable = 'off';

handles.Panel2.Visible = 'off';
handles.Panel3.Visible = 'off';

handles.Width_Input.String = '20';
handles.ViewWidth = 20;

handles.Channel_Selector.Value = 1;
handles.IsoSurfaces = [];
handles.fh = -1;
handles.ah = -1;

handles.ShowIsoSurfaces_Check.Value = 1;
handles.nIsoValues_Input.String = '3';
handles.nIso = 3;
handles.IsoValueLow_Input.String = '20';
handles.IsoLow = 20;
handles.IsoValueHigh_Input.String = '200';
handles.IsoHigh = 200;
handles.isoValues = round(linspace(handles.IsoHigh, handles.IsoLow, handles.nIso)');

ColorMaps = [{'parula'},{'jet'},{'cool'},...
             {'hot'},{'winter'},{'spring'},...
             {'summer'},{'bone'},{'gray'},...
             {'copper'},{'pink'},{'autumn'}];



handles.ColorMap = cool(100);
set(handles.ColorMaps_ListBox,'String',ColorMaps,'Value',3)

handles.RelativeHeight_Input.String = '3';
handles.dZdXRatio = 1;
set(handles.FilterSize_Display,'String','Filter Size:')

handles.DeleteEdgeMarkers_Check.Value = 1;

handles.SmallStepSize_ListBox.String =  [{'0.5 micron'},{'1 micron'},{'1/10 of View Width'}];
handles.SmallStepSize_ListBox.Value = 1;

guidata(hObject, handles);

% UIWAIT makes abacus wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = abacus_OutputFcn(hObject, eventdata, handles)

    varargout{1} = handles.output;


% --- Executes on button press in Select_File_Button.
function Select_File_Button_Callback(hObject, eventdata, handles)

    [FileName,PathName,FilterIndex] = uigetfile('*.tif');

    % Popup window to enter pixel size and z-step, but for now hard-code it.

    if ~isequal(0,FileName)

            [PixelSize, ZStep] = uiGetDataDimensions;
```

```
            if ~isequal(0,PixelSize) && ~isequal(0,ZStep)

                    handles.dx = PixelSize; % microns (pixel size - same as dy)
                    handles.dz = ZStep;     % microns
                    handles.dZdXRatio = handles.dz/handles.dx;

                    handles.ImageFileName = fullfile(PathName,FileName);
                    info = imfinfo(handles.ImageFileName);
                    handles.nZ = size(info,1);
                    handles.ImageStackSize = [info(1).Height, info(1).Width, handles.nZ];
                    set(handles.ImageName_Display,'String',FileName)

                    handles.Voxel_Dimension_Display.String = [{['pixel = ' ,sprintf('%#1.3f',PixelSize) ,' um']};...
                                                              {['zstep = ' ,sprintf('%#1.3f',ZStep) ,' um']}];

                    handles.Voxel_Dimension_Display.Enable = 'on';
                    handles.XYSigmaGaussFilt_Input.Enable  = 'on';
                    handles.ZSigmaGaussFilt_Input.Enable   = 'on';

                    handles.XYSigmaGaussFilt_Input.String  = '1.20';
                    %-------------------------------------------------------------------------------
                    xySigma = str2double(handles.XYSigmaGaussFilt_Input.String);
                    zSigma  = xySigma/handles.dZdXRatio;
                    handles.ZSigmaGaussFilt_Input.String = sprintf('%0.2f', zSigma);
                        nP = num2str(2*ceil(2*xySigma)+1);
                        nZ = num2str(2*ceil(2*zSigma )+1);
                    set(handles.FilterSize_Display,'String',['Filter Size:   ',nP,' x ',nP,' x ',nZ])
                    %-------------------------------------------------------------------------------

                    handles.text4.Enable = 'on';
                    handles.text5.Enable = 'on';
                    handles.text6.Enable = 'on';
                    handles.LoadAndProcessData_Button.Enable = 'on';

                    guidata(hObject, handles);
            end
    end


%-------------------------------------------------------------------------------------------------------
function handles = AdjustAxisTickLabels(handles)

    XLim = handles.ah.XLim;

    handles.ah.XTick = [XLim(1),...
                        handles.X_Position_Slider.Value,...
                        XLim(end)];

    handles.ah.XTickLabel = [{sprintf('%0.1f',XLim(1))};...
                             {sprintf('%0.1f',handles.X_Position_Slider.Value)};...
                             {sprintf('%0.1f',XLim(end))}];

%-------------------------------------------------------------------------------------------------------
function handles = Update_IsoSurfaces(handles)

        handles.isoValues = round(linspace(handles.IsoHigh, handles.IsoLow, handles.nIso)');

        if ~isgraphics(handles.fh)
            % Create Figure if none exists ---------------------------------------------------------
                handles.fh = figure;
                ScreenSize = get(0,'ScreenSize');  W = 1000; H = 700;
                set(handles.fh,'Color',[1 1 1],'NumberTitle','off','ToolBar','none','MenuBar','none',…
                                'Name','Abacus Rendering Window',...
                                'Position',[(ScreenSize(3)/2 - W/2) (ScreenSize(4)/2 - H/2) W H])

            % Create Axes in Figure, set axis limits and various params ----------------------------
                handles.ah = axes(handles.fh);
                handles.ah.Position = [0.10, 0.110, 0.7750, 0.8150];

                set(handles.ah,'CameraViewAngleMode','manual','CameraViewAngle',7.5,'BoxStyle','full',...
                               'DataAspectRatio', [1 1 1],'DataAspectRatioMode','manual',...
                               'XLim',handles.AxisSubLimits(1:2),...
                               'YLim',handles.AxisSubLimits(3:4),...
                               'ZLim',handles.AxisSubLimits(5:6));

                xlabel(handles.ah,'X (\mum)');
                ylabel(handles.ah,'Y (\mum)');
                zlabel(handles.ah,'Z (\mum)');

                view(handles.ah,60,15)
                rotate3d(handles.ah,'on')
                box(handles.ah,'on')
                grid(handles.ah,'on')
                camproj(handles.ah,'perspective')

                set(handles.ViewAngleZoom_Slider,'Min',5,'Max',12,'Value',7.5,'Enable','on','SliderStep',[0.1/7 ,1/7])
                set(handles.ViewZoom_Display,'String',['Angle Zoom ',sprintf('%0.1f',handles.ViewAngleZoom_Slider.Value)])

            % Create Colorbar ----------------------------------------------------------------------
```

```
            handles.HC = colorbar(handles.ah);
            set(handles.HC,'Position',[0.93 0.780 0.020 0.18],'LineWidth',1,...
                           'TickLength',0.04,'FontSize',8,'TickDirection','both');
            colormap(handles.ah, handles.ColorMap)
            caxis(handles.ah, [handles.IsoLow, handles.IsoHigh])
            handles.HC.Ticks = flipud(handles.isoValues)'; %linspace(handles.MagThresh, handles.MagHigh, 12);
            handles.HC.TickLabels = num2cell(flipud(handles.isoValues)); %num2cell(round(handles.HC.Ticks));

            obj = findall(handles.ah,'Type','Scatter');
            delete(obj)
            handles.ShowCellLocations_Check.Value = 0;
            handles.CellCount_Display.String = 'nCells: 0';

        end

        Alpha    = [linspace(0.3,0.1,handles.nIso)'];
        Cmap     = flipud(handles.ColorMap);
        Cind     = round(linspace(1,size(Cmap,1),handles.nIso))';
        %-----------------------------------------------------------------------
            OldIsoSurfaces  = findall(handles.ah,'Type','Patch'); % Grab old IsoSurfaces and delete them
            OldCrossSection = findall(handles.ah,'Type','Surface');
            OldLines        = findall(handles.ah,'Type','Line');
        %-----------------------------------------------------------------------
            handles.ah.XColor = [0.8 0.8 0.8];
            handles.ah.YColor = [0.8 0.8 0.8];
            handles.ah.ZColor = [0.8 0.8 0.8];
            drawnow
         %-----------------------------------------------------------------------
            handles.IsoSurfaces = cell(handles.nIso,1);
            Ch = handles.Channel_Selector.Value;
         % ----------------------------------------------------------------------
            handles.ImageSubVolume = single(squeeze(handles.RGBsmooth(:,handles.XsubIndices,:,Ch)));
         %-----------------------------------------------------------------------

        % Create New IsoSurfaces ------------------------------------------------
        if logical(handles.ShowIsoSurfaces_Check.Value)
            hold(handles.ah,'on')
            for n = 1:handles.nIso
                s1 = isosurface(handles.XsubMesh,...
                                handles.YsubMesh,...
                                handles.ZsubMesh,...
                                handles.ImageSubVolume,...
                                handles.isoValues(n,1));

                IsoSurfaces{n,1} = patch(handles.ah, s1);
                IsoSurfaces{n,1}.EdgeColor = 'none';
                IsoSurfaces{n,1}.FaceColor = Cmap(Cind(n,1),:);
                IsoSurfaces{n,1}.FaceAlpha = Alpha(n,1);
            end
            hold(handles.ah,'off')
        end
        % Create New YZ Cross-Section -------------------------------------------
            handles = Update_YZ_Cross_Section(handles);
        %-----------------------------------------------------------------------

        LightObjs = findall(handles.ah,'Type','Light');
        if length(LightObjs) < 2
            camlight(handles.ah, 90,0,'local');
            camlight(handles.ah,-90,0,'local');
        elseif length(LightObjs) > 4
            delete(LightObjs)
        end
        % Delete Old Graphics Objects -----------------------------
            delete(OldIsoSurfaces)
        %-----------------------------------------------------------

        set(handles.ah,'XLim',handles.AxisSubLimits(1:2),...
                       'YLim',handles.AxisSubLimits(3:4),...
                       'ZLim',handles.AxisSubLimits(5:6));

        handles = AdjustAxisTickLabels(handles);
            handles.ah.XColor = [0.1500 0.1500 0.1500];
            handles.ah.YColor = [0.1500 0.1500 0.1500];
            handles.ah.ZColor = [0.1500 0.1500 0.1500];

        drawnow
%-------------------------------------------------------------------------------------------------------------
function handles = Update_YZ_Cross_Section(handles)

            OldCrossSection = findall(handles.ah,'Type','Surface');
            OldLines        = findall(handles.ah,'Type','Line');
            delete(OldCrossSection)
            delete(OldLines)

            if logical(handles.ShowYZCrossSection_Button.Value)
                hold(handles.ah,'on')
                    Xpos = handles.X_Position_Slider.Value;
                    handles.YZSlice = slice(handles.ah,handles.XsubMesh,...
                                                       handles.YsubMesh,...
                                                       handles.ZsubMesh,...
```

```
                                                handles.ImageSubVolume, Xpos, [], []);

                    handles.YZSlice.FaceAlpha = 0.85;
                    handles.YZSlice.EdgeColor = 'none';
                    handles.YZSlice.AmbientStrength  = 1;
                    handles.YZSlice.DiffuseStrength  = 0;
                    handles.YZSlice.SpecularStrength = 0;
                    %handles.YZSlice.CData(handles.YZSlice.CData < handles.IsoLow) = NaN;
                end

                caxis(handles.ah, [handles.IsoLow, handles.IsoHigh])
                hold(handles.ah,'off')


%-------------------------------------------------------------------------------------------------------
function handles = UpdateCellCount(handles)

        Ch = handles.Channel_Selector.Value;
        Volume = squeeze(handles.RGBsmooth(:,:,:,Ch));
        H = str2double(handles.RelativeHeight_Input.String);
        M3D = imextendedmax(Volume,H);
        [L, NUM] = bwlabeln(M3D);
        stats = regionprops(L,'Centroid'); % Grab centers of 3d-objects
        C = struct2cell(stats);
        C = C';
        A = cell2mat(C);
        % Filter Cells by thresholding by brightness ----------------------------------------
            Ind = floor(A);
            Magnitude = zeros(length(A),1);
            for k = 1:length(A)
                Magnitude(k,1) = Volume(Ind(k,2),Ind(k,1),Ind(k,3));
            end
            A = A(Magnitude >= handles.IsoLow,:);  % Positions are in units of pixel indices
        %-----------------------------------------------------------------------------------
        if logical(handles.DeleteEdgeMarkers_Check.Value)
            w = 1;
            [ii,jj,kk] = size(L);
            ind1 = find( A(:,1) <= (min(A(:,1)) + w)  |  A(:,1) >= (max(A(:,1)) - w) );
            ind2 = find( A(:,2) <= (min(A(:,2)) + w)  |  A(:,2) >= (max(A(:,2)) - w) );
            ind3 = find( A(:,3) <= (min(A(:,3)) + w)  |  A(:,3) >= (max(A(:,3)) - w) );
            idx = [ind1; ind2; ind3];
            A(idx,:) = [];
        end
        %-----------------------------------------------------------------------------------
            handles.CellCountLocationsVoxels = [A(:,1), A(:,2), A(:,3)];
            X = handles.dx*A(:,1) - handles.dx;
            Y = handles.dx*A(:,2) - handles.dx;
            Z = handles.dz*A(:,3) - handles.dz;
            handles.CellCountLocationsMicrons = [X, Y, Z];
        %-----------------------------------------------------------------------------------
            obj = findall(handles.ah,'Type','Scatter');
            delete(obj)
        %-----------------------------------------------------------------------------------
            handles.CellCount = size(A,1);
            set(handles.CellCount_Display,'String',['nCells: ' num2str(handles.CellCount)])
            hold(handles.ah,'on')
                scatter3(handles.ah, X, Y, Z, 14,...
                         'MarkerEdgeColor', 'k',...
                         'MarkerFaceColor', [1, 0, 0]);
            hold(handles.ah,'off')

%-------------------------------------------------------------------------------------------------------
function FP = WaitToProcessDialogue

        ScreenSize = get(0,'ScreenSize');
        FP = figure;
        W = 260;
        H = 75;
        set(FP,'ToolBar','none','MenuBar','none',...
              'NumberTitle','off','Name','Abacus - wait',...
              'Position',[(ScreenSize(3)/2 - W/2) (ScreenSize(4)/2 - H/2) W H])
        a = gca;
        set(a,'Visible','off','XLim',[0 1],'YLim',[0 1])
        text(a,0,0.5,[{'Counting Cells'}],'Color',[0.4 0.4 0.4],...
            'FontSize',16,'FontName','Monospaced','FontWeight','bold')
        drawnow

%-------------------------------------------------------------------------------------------------------
function Apply_Saturation_CheckBox_Callback(hObject, eventdata, handles)



%-------------------------------------------------------------------------------------------------------
function XYSigmaGaussFilt_Input_Callback(hObject, eventdata, handles)

        xySigma = str2double(handles.XYSigmaGaussFilt_Input.String);

        if ~isnan(xySigma)
            if xySigma < 0.01 | xySigma > 10
                xySigma = 0.5;
```

```
            end
            set(handles.XYSigmaGaussFilt_Input,'String',sprintf('%0.2f',xySigma))
            zSigma  = str2double(handles.ZSigmaGaussFilt_Input.String);
            nP = num2str(2*ceil(2*xySigma)+1);
            nZ = num2str(2*ceil(2*zSigma )+1);
            set(handles.FilterSize_Display,'String',['Filter Size:   ',nP,' x ',nP,' x ',nZ])
        end

        guidata(hObject, handles);

%--------------------------------------------------------------------------------------------------------
function XYSigmaGaussFilt_Input_CreateFcn(hObject, eventdata, handles)

    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

%--------------------------------------------------------------------------------------------------------
function Xsize_BoxFilt_Input_Callback(hObject, eventdata, handles)



%--------------------------------------------------------------------------------------------------------
function Xsize_BoxFilt_Input_CreateFcn(hObject, eventdata, handles)

    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end


%--------------------------------------------------------------------------------------------------------
function ZSigmaGaussFilt_Input_Callback(hObject, eventdata, handles)

        zSigma = str2double(handles.ZSigmaGaussFilt_Input.String);
        xySigma  = str2double(handles.XYSigmaGaussFilt_Input.String);

        if ~isnan(zSigma)
            if zSigma < 0.01 | zSigma > 10
                zSigma = xySigma/handles.dZdXRatio;;
            end
            set(handles.ZSigmaGaussFilt_Input,'String',sprintf('%0.2f',zSigma))
            nP = num2str( 2*ceil(2*xySigma) + 1 );
            nZ = num2str( 2*ceil(2*zSigma ) + 1 );
            set(handles.FilterSize_Display,'String',['Filter Size:   ',nP,' x ',nP,' x ',nZ])
        end

        guidata(hObject, handles);


%--------------------------------------------------------------------------------------------------------
function ZSigmaGaussFilt_Input_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end


%--------------------------------------------------------------------------------------------------------
function LoadAndProcessData_Button_Callback(hObject, eventdata, handles)

            ImageSaturate = @(Im,percent) imadjust(Im,stretchlim(Im,[percent/100, (100-percent)/100]));

            % Read-in and Process images ------------------------------------------------------------
                handles.RGBraw = zeros([handles.ImageStackSize,3], 'uint8');

                h = waitbar(0,'Loading Data');

                for z = 1:handles.nZ
                    waitbar(z/handles.nZ,h);
                    drawnow
                    IM = imread(handles.ImageFileName,z);
                    if isequal(size(IM,3),3); % If Image is RGB----------------------------
                        if logical(handles.Apply_Saturation_CheckBox.Value)
                            handles.RGBraw(:,:,z,1) = ImageSaturate(squeeze(IM(:,:,1)), 1);
                            handles.RGBraw(:,:,z,2) = ImageSaturate(squeeze(IM(:,:,2)), 1);
                            handles.RGBraw(:,:,z,3) = ImageSaturate(squeeze(IM(:,:,3)), 1);
                        else
                            handles.RGBraw(:,:,z,1) = squeeze(IM(:,:,1));
                            handles.RGBraw(:,:,z,2) = squeeze(IM(:,:,2));
                            handles.RGBraw(:,:,z,3) = squeeze(IM(:,:,3));
                        end
                        handles.Channel_Selector.Enable = 'on';
                        handles.Channel_Selector.Value = 1;
                    else % If image is grayscale ----------------------------------------------
                        if logical(handles.Apply_Saturation_CheckBox.Value)
                            handles.RGBraw(:,:,z,1) = ImageSaturate(IM, 1);
                        else
                            handles.RGBraw(:,:,z,1) = squeeze(IM);
                        end
                        handles.Channel_Selector.Enable = 'off';
```

```
                %handles.Channel_Selector.Value = 1;
                %-------------------------------------------------------------------
            end
        end

    % Apply Gaussian smoothing ------------------------------------------------------------------------------------

        zSigma = str2double(handles.ZSigmaGaussFilt_Input.String);
        xySigma  = str2double(handles.XYSigmaGaussFilt_Input.String);

        waitbar(1,h,['Applying 3D Gaussian Smoothing'])

        handles.RGBsmooth = zeros([handles.ImageStackSize,3], 'uint8');

        if isequal(size(IM,3),3)
            handles.RGBsmooth(:,:,:,1) = imgaussfilt3(handles.RGBraw(:,:,:,1),...
                                                [xySigma,xySigma,zSigma], 'padding', 'circular');
            handles.RGBsmooth(:,:,:,2) = imgaussfilt3(handles.RGBraw(:,:,:,2),...
                                                [xySigma,xySigma,zSigma], 'padding', 'circular');
            handles.RGBsmooth(:,:,:,3) = imgaussfilt3(handles.RGBraw(:,:,:,3),...
                                                [xySigma,xySigma,zSigma], 'padding', 'circular');
        else
            handles.RGBsmooth(:,:,:,1) = imgaussfilt3(handles.RGBraw(:,:,:,1),...
                                                [xySigma,xySigma,zSigma], 'padding', 'circular');
        end

        pause(0.25)
        try
            delete(h);
        end
    %-------------------------------------------------------------------------------------------------------------

        handles.X_Axis_Vector = (handles.dx*[0:handles.ImageStackSize(2)-1])';
        handles.Y_Axis_Vector = (handles.dx*[0:handles.ImageStackSize(1)-1])';
        handles.Z_Axis_Vector = (handles.dz*[0:handles.nZ-1])';

    % Setup Slider and Value Display-------------------------------------------------------------------------------

        handles.PositionSliderMinValue_Display.String = sprintf('%#0.1f',handles.ViewWidth/2);
        handles.PositionSliderMaxValue_Display.String = ...
                                    sprintf('%#0.1f',handles.X_Axis_Vector(end)-handles.ViewWidth/2);

        Range = handles.X_Axis_Vector(end)-handles.ViewWidth;
        %SmallStep = 0.1*handles.ViewWidth/Range;
        SmallStep = 0.5/Range;
        BigStep = handles.ViewWidth/Range;

        set(handles.X_Position_Slider,'Value',handles.X_Axis_Vector(end)/2,'SliderStep',[SmallStep,BigStep],...
                               'Min',handles.ViewWidth/2,'Max',handles.X_Axis_Vector(end)-handles.ViewWidth/2)

        set(handles.SliderBarPosition_Display,'String',['X Position: ',...
                                         sprintf('%0.1f',handles.X_Axis_Vector(end)/2),' um'])

    % Create MESH grid ---------------------------------------------------------------------------

        [handles.Xmesh, handles.Ymesh, handles.Zmesh] = meshgrid(handles.X_Axis_Vector,...
                                                                 handles.Y_Axis_Vector,...
                                                                 handles.Z_Axis_Vector);

    % Calculate Sub-Mesh, Sub-Image, and Sub-Indices ---------------------------------------------------
        handles = Update_SubVolume_And_AxisLimits(handles);
        handles = Update_IsoSurfaces(handles);

        obj = findall(handles.ah,'Type','Scatter');    % Find old cell markers if they exist and delete them
        delete(obj)
        handles.CellCount_Display.String = 'nCells: 0';
    %-------------------------------------------------------------------------------------------------------
        handles.Panel2.Visible = 'on';
        handles.Panel3.Visible = 'on';
        % If Box is checked, recalculate cell count --------------
        if logical(handles.ShowCellLocations_Check.Value)
            FP = WaitToProcessDialogue; drawnow;
            handles = UpdateCellCount(handles);
            delete(FP)
        end
    %-----------------------------------------------------------

        guidata(hObject, handles);

%-----------------------------------------------------------------------------------------------------------
function handles = Update_SubVolume_And_AxisLimits(handles)

    Xpos = handles.X_Position_Slider.Value;
    Width = handles.ViewWidth;
    X_Axis_Vector = handles.X_Axis_Vector;

    handles.XsubIndices = find( abs(X_Axis_Vector - Xpos) <= (Width/2 + handles.dx/2) );

    handles.XsubMesh = handles.Xmesh( :, handles.XsubIndices, : );
    handles.YsubMesh = handles.Ymesh( :, handles.XsubIndices, : );
```

```
         handles.ZsubMesh = handles.Zmesh( :, handles.XsubIndices, : );

         handles.AxisSubLimits = [handles.XsubMesh(1,1,1),handles.XsubMesh(1,end,1),...
                                  handles.YsubMesh(1,1,1),handles.YsubMesh(end,1,1),...
                                  handles.ZsubMesh(1,1,1),handles.ZsubMesh(1,1,end)];

%--------------------------------------------------------------------------------------------------------
function X_Position_Slider_Callback(hObject, eventdata, handles)

        set(handles.SliderBarPosition_Display, 'String', ['X Position: ',...
                    sprintf('%0.2f',handles.X_Position_Slider.Value),' um'])

        handles = Update_SubVolume_And_AxisLimits(handles);
        handles = Update_IsoSurfaces(handles);

        guidata(hObject, handles);

%--------------------------------------------------------------------------------------------------------
function X_Position_Slider_CreateFcn(hObject, eventdata, handles)

        if isequal(get(hObject,'BackgroundColor'),...
                   get(0,'defaultUicontrolBackgroundColor'))
                   set(hObject,'BackgroundColor',[.9 .9 .9]);
        end

%--------------------------------------------------------------------------------------------------------
function Width_Input_Callback(hObject, eventdata, handles)

        val = round(str2num(handles.Width_Input.String));

        if isempty(val)
            handles.Width_Input.String = handles.ViewWidth;
        else
            handles.ViewWidth = val;
            handles.Width_Input.String = num2str(val);

            Range = handles.X_Axis_Vector(end) - handles.ViewWidth;

            V = handles.SmallStepSize_ListBox.Value;
            Range = handles.X_Axis_Vector(end)-handles.ViewWidth;
            BigStep = handles.ViewWidth/Range;

            switch V
                case 1
                        SmallStep = 0.5/Range;
                case 2
                        SmallStep = 1/Range;
                case 3
                        SmallStep = 0.1*BigStep;
            end

            set(handles.X_Position_Slider,'SliderStep',[SmallStep,BigStep])

            handles.PositionSliderMinValue_Display.String = sprintf('%#0.1f', handles.ViewWidth/2);
            handles.PositionSliderMaxValue_Display.String = ...
                                              sprintf('%#0.1f', handles.X_Axis_Vector(end) - handles.ViewWidth/2);

            set(handles.X_Position_Slider, 'Value', handles.X_Axis_Vector(end)/2,...
                                           'SliderStep', [SmallStep,BigStep],...
                                           'Min', handles.ViewWidth/2,...
                                           'Max', handles.X_Axis_Vector(end) - handles.ViewWidth/2)

            set(handles.SliderBarPosition_Display, 'String',...
                                             ['X Position: ', sprintf('%0.1f', handles.X_Axis_Vector(end)/2),' um'])
            %-----------------------------------------------------------------------------------------------------
            handles = Update_SubVolume_And_AxisLimits(handles);
            handles = Update_IsoSurfaces(handles);
        end

        guidata(hObject, handles);

%--------------------------------------------------------------------------------------------------------
function Width_Input_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

%--------------------------------------------------------------------------------------------------------
function [PixelSize, ZStep] = uiGetDataDimensions

        global PixelSize ZStep

        R = groot;
        SS = R.ScreenSize(1,3:4);

        PixelSize = 0;
        ZStep = 0;

        d = dialog('NumberTitle','off','Position',[SS(1)/2-160, SS(2)/2-100, 320, 250],'Name','Abacus - Dimensions Input');
```

```
        txt1  = uicontrol('Parent',d,'Style','text','Position',[27 200 185 21],'Units','pixels',...
                         'String','Enter Data Dimensions:','FontSize',12,'FontWeight','bold','HorizontalAlignment','left');

        txt2  = uicontrol('Parent',d,'Style','text','Position',[49 153 127 18],'Units','pixels',...
                         'String','Pixel Width (um):','FontSize',11,'FontWeight','normal','HorizontalAlignment','right');

        txt3  = uicontrol('Parent',d,'Style','text','Position',[49 112 127 18],'Units','pixels',...
                         'String','Z-Step (um):','FontSize',11,'FontWeight','normal','HorizontalAlignment','right');

        inpt1 = uicontrol('Parent',d,'Style','edit','Position',[183 146 68 29],'Units','pixels',...
                         'String','0.593','FontSize',11,'FontWeight','normal','HorizontalAlignment','center');

        inpt2 = uicontrol('Parent',d,'Style','edit','Position',[183 105 68 29],'Units','pixels',...
                         'String','2.100','FontSize',11,'FontWeight','normal','HorizontalAlignment','center');

        btn1  = uicontrol('Parent',d,'Style','pushbutton','Position',[96 36 136 38],'Units','pixels',...
                         'String','OK','FontSize',12,'FontWeight','normal','HorizontalAlignment','center',...
                         'Callback',{@GrabValues,inpt1,inpt2});

        uiwait(d);

        %-----------------------------------------
        function GrabValues(btn1,event,inpt1,inpt2)

            global PixelSize ZStep

            PixelSize = str2num(inpt1.String);
                if isempty(PixelSize); PixelSize = 0; end
            ZStep = str2num(inpt2.String);
                if isempty(ZStep); ZStep = 0; end

            delete(gcf)

%-------------------------------------------------------------------------------------------------
function Channel_Selector_Callback(hObject, eventdata, handles)

        handles = Update_IsoSurfaces(handles);

        obj = findall(handles.ah,'Type','Scatter');  % Find old cell markers if they exist and delete them
        delete(obj)
        handles.CellCount_Display.String = 'nCells: 0';

        % If Box is checked, recalculate cell count --------------------------
        if logical(handles.ShowCellLocations_Check.Value)
            FP = WaitToProcessDialogue; drawnow;
            handles = UpdateCellCount(handles);
            delete(FP)
        end
        %---------------------------------------------------------------------

        guidata(hObject, handles);

%-------------------------------------------------------------------------------------------------
function Channel_Selector_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end


%-------------------------------------------------------------------------------------------------
function IsoValueLow_Input_Callback(hObject, eventdata, handles)

        val = str2double(handles.IsoValueLow_Input.String);

        if ~isnan(val)
            handles.IsoLow = val;
            colormap(handles.ah, handles.ColorMap)
            caxis(handles.ah, [handles.IsoLow, handles.IsoHigh])
            handles.isoValues = round(linspace(handles.IsoHigh, handles.IsoLow, handles.nIso)');
            handles.HC.Ticks = flipud(handles.isoValues)';
            handles.HC.TickLabels = num2cell(flipud(handles.isoValues));
            handles = Update_IsoSurfaces(handles);

            obj = findall(handles.ah,'Type','Scatter');
            delete(obj)
            handles.CellCount_Display.String = 'nCells: 0';

            % If Box is checked, recalculate cell count --------------
            if logical(handles.ShowCellLocations_Check.Value)
                FP = WaitToProcessDialogue; drawnow;
                handles = UpdateCellCount(handles);
                delete(FP)
            end
            %-------------------------------------------------------
            guidata(hObject, handles);

        end
```

```
%--------------------------------------------------------------------------------------------
function IsoValueLow_Input_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

%--------------------------------------------------------------------------------------------
function IsoValueHigh_Input_Callback(hObject, eventdata, handles)

        val = str2double(handles.IsoValueHigh_Input.String);

        if ~isnan(val)
            handles.IsoHigh = val;
            colormap(handles.ah, handles.ColorMap)
            caxis(handles.ah, [handles.IsoLow, handles.IsoHigh])
            handles.isoValues = round(linspace(handles.IsoHigh, handles.IsoLow, handles.nIso)');
            handles.HC.Ticks = flipud(handles.isoValues)';
            handles.HC.TickLabels = num2cell(flipud(handles.isoValues));

            handles = Update_IsoSurfaces(handles);

            guidata(hObject, handles);
        end

%--------------------------------------------------------------------------------------------
function IsoValueHigh_Input_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

%--------------------------------------------------------------------------------------------
function nIsoValues_Input_Callback(hObject, eventdata, handles)

        val = str2double(handles.nIsoValues_Input.String);

        if ~isnan(val)
            handles.nIso = val;
            colormap(handles.ah, handles.ColorMap)
            caxis(handles.ah, [handles.IsoLow, handles.IsoHigh])
            handles.isoValues = round(linspace(handles.IsoHigh, handles.IsoLow, handles.nIso)');
            handles.HC.Ticks = flipud(handles.isoValues)';
            handles.HC.TickLabels = num2cell(flipud(handles.isoValues));

            handles = Update_IsoSurfaces(handles);

            guidata(hObject, handles);
        end

%--------------------------------------------------------------------------------------------
function nIsoValues_Input_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

%--------------------------------------------------------------------------------------------
function ColorMaps_ListBox_Callback(hObject, eventdata, handles)

        handles.ColorMap = eval([handles.ColorMaps_ListBox.String{handles.ColorMaps_ListBox.Value},'(100)']);
        colormap(handles.ah, handles.ColorMap)
        caxis(handles.ah, [handles.IsoLow, handles.IsoHigh])
        handles.isoValues = round(linspace(handles.IsoHigh, handles.IsoLow, handles.nIso)');
        handles.HC.Ticks = flipud(handles.isoValues)';
        handles.HC.TickLabels = num2cell(flipud(handles.isoValues));

        handles = Update_IsoSurfaces(handles);

        guidata(hObject, handles);

%--------------------------------------------------------------------------------------------
function ColorMaps_ListBox_CreateFcn(hObject, eventdata, handles)

        if ispc && isequal(get(hObject,'BackgroundColor'), ...
                get(0,'defaultUicontrolBackgroundColor'))
            set(hObject,'BackgroundColor','white');
        end

%--------------------------------------------------------------------------------------------
function ShowCellLocations_Check_Callback(hObject, eventdata, handles)

        % If Box is checked, recalculate cell count --------------
        if logical(handles.ShowCellLocations_Check.Value)
            FP = WaitToProcessDialogue; drawnow;
            handles = UpdateCellCount(handles);
            delete(FP)
        else
            delete(findall(handles.ah,'Type','Scatter'))
            set(handles.CellCount_Display,'String',['nCells: 0'])
```

```
        end
        %-----------------------------------------------------
        guidata(hObject, handles);

%-------------------------------------------------------------------------------------------
function ViewAngleZoom_Slider_Callback(hObject, eventdata, handles)

        val = round( get(handles.ViewAngleZoom_Slider,'Value')*10)/10;
        set(handles.ah,'CameraViewAngle',val)
        set(handles.ViewAngleZoom_Slider,'Value',val)
        set(handles.ViewZoom_Display,'String',['Angle Zoom ',sprintf('%0.1f',val)])

        guidata(hObject, handles);

%-------------------------------------------------------------------------------------------
function ViewAngleZoom_Slider_CreateFcn(hObject, eventdata, handles)

    if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
    end

%-------------------------------------------------------------------------------------------
function RelativeHeight_Input_Callback(hObject, eventdata, handles)

        val = str2double(handles.RelativeHeight_Input.String);

        if ~isnan(val)
            val = min([val;200]);
            handles.RelativeHeight_Input.String = num2str(val);
        else
            handles.RelativeHeight_Input.String = '3';
        end

        % If Box is checked, recalculate cell count --------------
        if logical(handles.ShowCellLocations_Check.Value)
            FP = WaitToProcessDialogue; drawnow;
            handles = UpdateCellCount(handles);
            delete(FP)
        end
        %-----------------------------------------------------
        guidata(hObject, handles);

%-------------------------------------------------------------------------------------------
function RelativeHeight_Input_CreateFcn(hObject, eventdata, handles)

    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

%-------------------------------------------------------------------------------------------
function DeleteEdgeMarkers_Check_Callback(hObject, eventdata, handles)


%-------------------------------------------------------------------------------------------
function Suggested_Z_Sigma_Button_Callback(hObject, eventdata, handles)

    xySigma = str2double(handles.XYSigmaGaussFilt_Input.String);
    zSigma  = xySigma/handles.dZdXRatio;
    handles.ZSigmaGaussFilt_Input.String = sprintf('%0.2f', zSigma);
    nP = num2str(2*ceil(2*xySigma)+1);
    nZ = num2str(2*ceil(2*zSigma )+1);
    set(handles.FilterSize_Display,'String',['Filter Size:   ',nP,' x ',nP,' x ',nZ])

    guidata(hObject, handles);

%-------------------------------------------------------------------------------------------
function ShowYZCrossSection_Button_Callback(hObject, eventdata, handles)

     % Create New YZ Cross-Section ------------------------------------------------
          handles = Update_YZ_Cross_Section(handles);
     %-----------------------------------------------------------------------------

     guidata(hObject, handles)

%-------------------------------------------------------------------------------------------
function ExportMarkers_Button_Callback(hObject, eventdata, handles)

    [FileName,PathName] = uiputfile('*.tif');

    if ~isequal(FileName,0)
        FilePath = fullfile(PathName,FileName);
        XYZ = round(handles.CellCountLocationsVoxels);
        XYZ(XYZ==0) = 1;
        XYZ = sortrows(XYZ,3);
        se = strel('disk',1);

        for z = 1:handles.ImageStackSize(1,3)
            idx = find(XYZ(:,3) == z)';
            ZSlice = zeros(handles.ImageStackSize(1,1:2),'uint8');
            for p = idx
```

```
                R = XYZ(p,2);
                C = XYZ(p,1);
                ZSlice(R,C) = 255;
            end
            ZSlice = imdilate(ZSlice,se);
            if z == 1
                imwrite(ZSlice,FilePath,'tif','WriteMode','overwrite','Compression','none')
            else
                imwrite(ZSlice,FilePath,'tif','WriteMode','append','Compression','none')
            end
        end
    end

%--------------------------------------------------------------------------------------------
function ShowIsoSurfaces_Check_Callback(hObject, eventdata, handles)

    handles = Update_IsoSurfaces(handles);

    guidata(hObject, handles)

%--------------------------------------------------------------------------------------------
function SmallStepSize_ListBox_Callback(hObject, eventdata, handles)

    V = handles.SmallStepSize_ListBox.Value;
    Range = handles.X_Axis_Vector(end)-handles.ViewWidth;
    BigStep = handles.ViewWidth/Range;

    switch V
        case 1
                SmallStep = 0.5/Range;
        case 2
                SmallStep = 1/Range;
        case 3
                SmallStep = 0.1*BigStep;
    end

    set(handles.X_Position_Slider,'SliderStep',[SmallStep,BigStep])

    guidata(hObject, handles)

%--------------------------------------------------------------------------------------------
function SmallStepSize_ListBox_CreateFcn(hObject, eventdata, handles)

    if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end


% --- Executes on button press in About_Button.
function About_Button_Callback(hObject, eventdata, handles)

   message = [{' Abacus written and designed by:  Keith R. Carney'};...
              {'     '};...
              {'         email: keith.carney@path.utah.edu '};...
              {'           or:     ny2ak2ut@hotmail.com '};...
              {'     '};...
              {'     '};...
              {'    University of Utah '};...
              {'    Department of Human Genetics '};...
              {'    Kristen M. Kwan Lab, 2018 '};...
              {'     '};
              {'     '}];

    f = figure('NumberTitle','off',...
               'Position',[200 200 440 200],...
               'Name','About Abacus',...
               'Resize','off',...
               'Toolbar','none',...
               'WindowStyle','normal',...
               'MenuBar','none');

    uicontrol('Parent',f,'Style','text','FontName','FixedWidth','FontWeight','bold',...
         'FontSize',10,'String',message,'Position',[15 5 440 170],...
         'HorizontalAlignment','Left');




% --- Creates and returns a handle to the GUI figure.
function h1 = abacus_LayoutFcn(policy)
% policy - create a new figure or use a singleton. 'new' or 'reuse'.

persistent hsingleton;
if strcmpi(policy, 'reuse') & ishandle(hsingleton)
    h1 = hsingleton;
    return;
```

```
end

appdata = [];
appdata.GUIDEOptions = struct(...
    'active_h', [], ...
    'taginfo', struct(...
    'figure', 2, ...
    'text', 26, ...
    'pushbutton', 7, ...
    'uipanel', 8, ...
    'checkbox', 5, ...
    'edit', 9, ...
    'slider', 3, ...
    'popupmenu', 4), ...
    'override', 0, ...
    'release', [], ...
    'resize', 'none', ...
    'accessibility', 'callback', ...
    'mfile', [], ...
    'callbacks', [], ...
    'singleton', [], ...
    'syscolorfig', [], ...
    'blocking', 0, ...
    'lastSavedFile', 'C:\Matlab Files\Abacus\Main Program Files\test\abacus.m', ...
    'lastFilename', 'C:\Matlab Files\Abacus\Main Program Files\abacus.fig');
appdata.lastValidTag = 'figure1';
appdata.GUIDELayoutEditor = [];
appdata.initTags = struct(...
    'handle', [], ...
    'tag', 'figure1');

h1 = figure(...
'Units',get(0,'defaultfigureUnits'),...
'Position',[680 548 325 620],...
'Visible',get(0,'defaultfigureVisible'),...
'Color',get(0,'defaultfigureColor'),...
'IntegerHandle','off',...
'MenuBar','none',...
'Name','abacus',...
'NumberTitle','off',...
'Tag','figure1',...
'Resize','off',...
'PaperPosition',get(0,'defaultfigurePaperPosition'),...
'ScreenPixelsPerInchMode','manual',...
'HandleVisibility','callback',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'Panel1';

h2 = uipanel(...
'Parent',h1,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'BorderType','beveledin',...
'BorderWidth',2,...
'Title',blanks(0),...
'Tag','Panel1',...
'Position',[-1 424 330 196],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'Select_File_Button';

h3 = uicontrol(...
'Parent',h2,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Select File',...
'Style',get(0,'defaultuicontrolStyle'),...
'Position',[5 134 90 56],...
'Callback',@(hObject,eventdata)abacus('Select_File_Button_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'Tag','Select_File_Button',...
'FontSize',10,...
'FontWeight','bold',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'uipanel1';

h4 = uipanel(...
'Parent',h2,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'Title',blanks(0),...
'Tag','uipanel1',...
'Position',[100 134 220 55],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

```
appdata = [];
appdata.lastValidTag = 'ImageName_Display';

h5 = uicontrol(...
'Parent',h4,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',blanks(0),...
'Style','text',...
'Position',[5 5 209 46],...
'Children',[],...
'ForegroundColor',[0 0.450980392156863 0.741176470588235],...
'Tag','ImageName_Display',...
'FontSize',9,...
'FontWeight','bold',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'Apply_Saturation_CheckBox';

h6 = uicontrol(...
'Parent',h2,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'ListboxTop',get(0,'defaultuicontrolListboxTop'),...
'Max',get(0,'defaultuicontrolMax'),...
'Min',get(0,'defaultuicontrolMin'),...
'SliderStep',get(0,'defaultuicontrolSliderStep'),...
'String','Apply 1% Pixel Saturation',...
'Style','checkbox',...
'Value',get(0,'defaultuicontrolValue'),...
'Position',[12 109 165 20],...
'BackgroundColor',get(0,'defaultuicontrolBackgroundColor'),...
'Callback',@(hObject,eventdata)abacus('Apply_Saturation_CheckBox_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ForegroundColor',get(0,'defaultuicontrolForegroundColor'),...
'Enable',get(0,'defaultuicontrolEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuicontrolVisible'),...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','Apply_Saturation_CheckBox',...
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuicontrolHandleVisibility'),...
'FontSize',9,...
'FontName',get(0,'defaultuicontrolFontName'),...
'FontAngle',get(0,'defaultuicontrolFontAngle'),...
'FontWeight',get(0,'defaultuicontrolFontWeight'));

appdata = [];
appdata.lastValidTag = 'Voxel_Dimension_Display';

h7 = uicontrol(...
'Parent',h2,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String',{  'pixel = 0.593 um'; 'zstep = 2.100 um' },...
'Style','text',...
'Position',[190 98 120 30],...
'Children',[],...
'Tag','Voxel_Dimension_Display',...
'FontSize',9,...
'FontName','Monospaced',...
'FontWeight','bold',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'LoadAndProcessData_Button';

h8 = uicontrol(...
'Parent',h2,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'ListboxTop',get(0,'defaultuicontrolListboxTop'),...
'Max',get(0,'defaultuicontrolMax'),...
'Min',get(0,'defaultuicontrolMin'),...
'SliderStep',get(0,'defaultuicontrolSliderStep'),...
'String','Load & Process',...
'Style',get(0,'defaultuicontrolStyle'),...
'Value',get(0,'defaultuicontrolValue'),...
'Position',[183 12 134 34],...
'BackgroundColor',[0 0.450980392156863 0.741176470588235],...
'Callback',@(hObject,eventdata)abacus('LoadAndProcessData_Button_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
```

```
'ForegroundColor',[1 1 0],...
'Enable',get(0,'defaultuicontrolEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuicontrolVisible'),...
'ButtonDownFcn',blanks(0),...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','LoadAndProcessData_Button',...
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuicontrolHandleVisibility'),...
'FontSize',11,...
'FontName',get(0,'defaultuicontrolFontName'),...
'FontAngle',get(0,'defaultuicontrolFontAngle'),...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'text23';

h9 = uicontrol(...
'Parent',h2,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','dz/dx = 3.54',...
'Style','text',...
'Position',[190 82 101 14],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text23',...
'FontSize',9,...
'FontName','Monospaced',...
'FontWeight',get(0,'defaultuicontrolFontWeight'));

appdata = [];
appdata.lastValidTag = 'uipanel7';

h10 = uipanel(...
'Parent',h2,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'Title','3D Gaussian Filter',...
'Tag','uipanel7',...
'Position',[10 9 165 87],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'XYSigmaGaussFilt_Input';

h11 = uicontrol(...
'Parent',h10,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'ListboxTop',get(0,'defaultuicontrolListboxTop'),...
'Max',get(0,'defaultuicontrolMax'),...
'Min',get(0,'defaultuicontrolMin'),...
'SliderStep',get(0,'defaultuicontrolSliderStep'),...
'String','0.5',...
'Style','edit',...
'Value',[],...
'Position',[59 28 40 23],...
'Callback',@(hObject,eventdata)abacus('XYSigmaGaussFilt_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ForegroundColor',get(0,'defaultuicontrolForegroundColor'),...
'Enable',get(0,'defaultuicontrolEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuicontrolVisible'),...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('XYSigmaGaussFilt_Input_CreateFcn',...
                                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','XYSigmaGaussFilt_Input',...
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuicontrolHandleVisibility'),...
'FontSize',10,...
'FontName',get(0,'defaultuicontrolFontName'),...
'FontAngle',get(0,'defaultuicontrolFontAngle'),...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'ZSigmaGaussFilt_Input';

h12 = uicontrol(...
```

```
'Parent',h10,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'ListboxTop',get(0,'defaultuicontrolListboxTop'),...
'Max',get(0,'defaultuicontrolMax'),...
'Min',get(0,'defaultuicontrolMin'),...
'SliderStep',get(0,'defaultuicontrolSliderStep'),...
'String','0.522',...
'Style','edit',...
'Value',[],...
'Position',[105 28 40 23],...
'Callback',@(hObject,eventdata)abacus('ZSigmaGaussFilt_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ForegroundColor',get(0,'defaultuicontrolForegroundColor'),...
'Enable',get(0,'defaultuicontrolEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuicontrolVisible'),...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('ZSigmaGaussFilt_Input_CreateFcn',...
                hObject,eventdata,guidata(hObject)), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','ZSigmaGaussFilt_Input',...
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuicontrolHandleVisibility'),...
'FontSize',10,...
'FontName',get(0,'defaultuicontrolFontName'),...
'FontAngle',get(0,'defaultuicontrolFontAngle'),...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'text22';

h13 = uicontrol(...
'Parent',h10,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','sigma:      xy           z',...
'Style','text',...
'Position',[16 54 120 14],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text22',...
'FontSize',9,...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'Suggested_Z_Sigma_Button';

h14 = uicontrol(...
'Parent',h10,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','sig z',...
'Style',get(0,'defaultuicontrolStyle'),...
'Position',[17 28 32 24],...
'Callback',@(hObject,eventdata)abacus('Suggested_Z_Sigma_Button_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'Tag','Suggested_Z_Sigma_Button',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'FilterSize_Display';

h15 = uicontrol(...
'Parent',h10,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'ListboxTop',get(0,'defaultuicontrolListboxTop'),...
'Max',get(0,'defaultuicontrolMax'),...
'Min',get(0,'defaultuicontrolMin'),...
'SliderStep',get(0,'defaultuicontrolSliderStep'),...
'String','Filter Size:   3 x 3 x 3',...
'Style','text',...
'Value',get(0,'defaultuicontrolValue'),...
'Position',[17 10 126 14],...
'BackgroundColor',get(0,'defaultuicontrolBackgroundColor'),...
'Callback',blanks(0),...
'Children',[],...
'ForegroundColor',get(0,'defaultuicontrolForegroundColor'),...
'Enable',get(0,'defaultuicontrolEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuicontrolVisible'),...
'ButtonDownFcn',blanks(0),...
```

```
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','FilterSize_Display',...
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuicontrolHandleVisibility'),...
'FontSize',get(0,'defaultuicontrolFontSize'),...
'FontName',get(0,'defaultuicontrolFontName'),...
'FontAngle',get(0,'defaultuicontrolFontAngle'),...
'FontWeight',get(0,'defaultuicontrolFontWeight'));

appdata = [];
appdata.lastValidTag = 'About_Button';

h16 = uicontrol(...
'Parent',h2,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','About Abacus',...
'Style',get(0,'defaultuicontrolStyle'),...
'Position',[237 52 81 20],...
'Callback',@(hObject,eventdata)abacus('About_Button_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'Tag','About_Button',...
'FontSize',7.5,...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'Panel2';

h17 = uipanel(...
'Parent',h1,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'BorderType','beveledin',...
'BorderWidth',2,...
'Title',blanks(0),...
'Tag','Panel2',...
'Position',[-4 149 330 262],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'X_Position_Slider';

h18 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',blanks(0),...
'Style','slider',...
'Position',[10 179 305 42],...
'BackgroundColor',[0.9 0.9 0.9],...
'Callback',@(hObject,eventdata)abacus('X_Position_Slider_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('X_Position_Slider_CreateFcn',...
                            hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','X_Position_Slider');

appdata = [];
appdata.lastValidTag = 'Width_Input';

h19 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','40',...
'Style','edit',...
'Position',[252 227 35 22],...
'Callback',@(hObject,eventdata)abacus('Width_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('Width_Input_CreateFcn',...
                              hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','Width_Input',...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'text7';

h20 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','um',...
'Style','text',...
'Position',[289 230 25 15],...
'Children',[],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
```

```
'Tag','text7',...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'text8';

h21 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','right',...
'String','View Width:',...
'Style','text',...
'Position',[177 230 73 15],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text8',...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'SliderBarPosition_Display';

h22 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','X Position: 150 um',...
'Style','text',...
'Position',[20 233 151 15],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','SliderBarPosition_Display',...
'FontSize',11,...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'PositionSliderMinValue_Display';

h23 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','20',...
'Style','text',...
'Position',[11 159 42 18],...
'Children',[],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'Tag','PositionSliderMinValue_Display',...
'FontSize',10);

appdata = [];
appdata.lastValidTag = 'PositionSliderMaxValue_Display';

h24 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','right',...
'String','0',...
'Style','text',...
'Position',[260 159 55 18],...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','PositionSliderMaxValue_Display',...
'FontSize',10);

appdata = [];
appdata.lastValidTag = 'Channel_Selector';

h25 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',{  'Ch 1'; 'Ch 2'; 'Ch 3' },...
'Style','popupmenu',...
'Value',[],...
'ValueMode',get(0,'defaultuicontrolValueMode'),...
'Position',[11 98 75 24],...
'Callback',@(hObject,eventdata)abacus('Channel_Selector_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('Channel_Selector_CreateFcn',...
```

```
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','Channel_Selector',...
'FontSize',10,...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'uipanel5';

h26 = uipanel(...
'Parent',h17,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'ForegroundColor',[0 0.450980392156863 0.741176470588235],...
'Title','IsoSurface Parameters',...
'Tag','uipanel5',...
'Position',[92 33 226 95],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'IsoValueLow_Input';

h27 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','90',...
'Style','edit',...
'Position',[143 12 35 22],...
'Callback',@(hObject,eventdata)abacus('IsoValueLow_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('IsoValueLow_Input_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','IsoValueLow_Input',...
'KeyPressFcn',blanks(0),...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'text14';

h28 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'String','Low',...
'Style','text',...
'Position',[144 35 33 15],...
'Children',[],...
'ForegroundColor',[0 0.450980392156863 0.741176470588235],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text14',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'IsoValueHigh_Input';

h29 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','200',...
'Style','edit',...
'Position',[181 12 35 22],...
'Callback',@(hObject,eventdata)abacus('IsoValueHigh_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('IsoValueHigh_Input_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','IsoValueHigh_Input',...
'KeyPressFcn',blanks(0),...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'text15';

h30 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'String','High',...
'Style','text',...
'Position',[182 35 33 15],...
```

```
'Children',[],...
'ForegroundColor',[0 0.450980392156863 0.741176470588235],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text15',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'text16';

h31 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'String','N',...
'Style','text',...
'Position',[109 35 21 15],...
'Children',[],...
'ForegroundColor',[0 0.450980392156863 0.741176470588235],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text16',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'nIsoValues_Input';

h32 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','3',...
'Style','edit',...
'Position',[106 12 28 22],...
'Callback',@(hObject,eventdata)abacus('nIsoValues_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('nIsoValues_Input_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','nIsoValues_Input',...
'KeyPressFcn',blanks(0),...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'ColorMaps_ListBox';

h33 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',blanks(0),...
'Style','popupmenu',...
'Value',[],...
'ValueMode',get(0,'defaultuicontrolValueMode'),...
'Position',[11 12 85 22],...
'Callback',@(hObject,eventdata)abacus('ColorMaps_ListBox_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('ColorMaps_ListBox_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','ColorMaps_ListBox',...
'FontSize',9);

appdata = [];
appdata.lastValidTag = 'text18';

h34 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment',get(0,'defaultuicontrolHorizontalAlignment'),...
'String','Color Map',...
'Style','text',...
'Position',[8 35 58 15],...
'Children',[],...
'ForegroundColor',[0 0.450980392156863 0.741176470588235],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','text18',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'ShowIsoSurfaces_Check';
```

```
h35 = uicontrol(...
'Parent',h26,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Show IsoSurfaces',...
'Style','checkbox',...
'Position',[12 57 115 20],...
'Callback',@(hObject,eventdata)abacus('ShowIsoSurfaces_Check_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','ShowIsoSurfaces_Check',...
'KeyPressFcn',blanks(0));

appdata = [];
appdata.lastValidTag = 'ViewAngleZoom_Slider';

h36 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String',{  'Slider' },...
'Style','slider',...
'Position',[10 56 75 23],...
'BackgroundColor',[0.9 0.9 0.9],...
'Callback',@(hObject,eventdata)abacus('ViewAngleZoom_Slider_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'BusyAction','cancel',...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('ViewAngleZoom_Slider_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','ViewAngleZoom_Slider');

appdata = [];
appdata.lastValidTag = 'ViewZoom_Display';

h37 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'ListboxTop',get(0,'defaultuicontrolListboxTop'),...
'Max',get(0,'defaultuicontrolMax'),...
'Min',get(0,'defaultuicontrolMin'),...
'SliderStep',get(0,'defaultuicontrolSliderStep'),...
'String','Angle Zoom: 10',...
'Style','text',...
'Value',get(0,'defaultuicontrolValue'),...
'Position',[9 35 80 14],...
'BackgroundColor',get(0,'defaultuicontrolBackgroundColor'),...
'Callback',blanks(0),...
'Children',[],...
'ForegroundColor',get(0,'defaultuicontrolForegroundColor'),...
'Enable',get(0,'defaultuicontrolEnable'),...
'TooltipString',blanks(0),...
'Visible',get(0,'defaultuicontrolVisible'),...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','ViewZoom_Display',...
'UserData',[],...
'KeyPressFcn',blanks(0),...
'KeyReleaseFcn',blanks(0),...
'HandleVisibility',get(0,'defaultuicontrolHandleVisibility'),...
'FontSize',get(0,'defaultuicontrolFontSize'),...
'FontName',get(0,'defaultuicontrolFontName'),...
'FontAngle',get(0,'defaultuicontrolFontAngle'),...
'FontWeight',get(0,'defaultuicontrolFontWeight'));

appdata = [];
appdata.lastValidTag = 'ShowYZCrossSection_Button';

h38 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Show YZ Cross Section',...
'Style','checkbox',...
'Position',[176 9 140 20],...
'Callback',@(hObject,eventdata)abacus('ShowYZCrossSection_Button_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'Tag','ShowYZCrossSection_Button');

appdata = [];
appdata.lastValidTag = 'SmallStepSize_ListBox';

h39 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
```

```
'Units',get(0,'defaultuicontrolUnits'),...
'String','1/10 of View Width',...
'Style','popupmenu',...
'Value',[],...
'ValueMode',get(0,'defaultuicontrolValueMode'),...
'Position',[141 146 124 22],...
'Callback',@(hObject,eventdata)abacus('SmallStepSize_ListBox_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('SmallStepSize_ListBox_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','SmallStepSize_ListBox');

appdata = [];
appdata.lastValidTag = 'text25';

h40 = uicontrol(...
'Parent',h17,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Slider Small Step:',...
'Style','text',...
'Position',[51 149 87 14],...
'Children',[],...
'Tag','text25',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'Panel3';

h41 = uipanel(...
'Parent',h1,...
'FontUnits',get(0,'defaultuipanelFontUnits'),...
'Units','pixels',...
'BorderType','beveledin',...
'BorderWidth',2,...
'Title',blanks(0),...
'Tag','Panel3',...
'Position',[-4 18 330 116],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'ShowCellLocations_Check';

h42 = uicontrol(...
'Parent',h41,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Calculate Cell Count',...
'Style','checkbox',...
'Position',[11 85 191 23],...
'Callback',@(hObject,eventdata)abacus('ShowCellLocations_Check_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'Tag','ShowCellLocations_Check',...
'FontSize',12,...
'FontWeight','bold');

appdata = [];
appdata.lastValidTag = 'CellCount_Display';

h43 = uicontrol(...
'Parent',h41,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','left',...
'String','nCells: 0',...
'Style','text',...
'Position',[9 16 142 22],...
'Children',[],...
'ForegroundColor',[0.63921568627451 0.0784313725490196 0.180392156862745],...
'Tag','CellCount_Display',...
'FontSize',14,...
'FontWeight','bold',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );

appdata = [];
appdata.lastValidTag = 'text21';

h44 = uicontrol(...
'Parent',h41,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'HorizontalAlignment','right',...
'String','ImExtendedMax Relative Height:',...
'Style','text',...
'Position',[23 46 160 14],...
'Children',[],...
'Tag','text21',...
'FontSize',get(0,'defaultuicontrolFontSize'),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );
```

```
appdata = [];
appdata.lastValidTag = 'RelativeHeight_Input';

h45 = uicontrol(...
'Parent',h41,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','10',...
'Style','edit',...
'Position',[184 44 25 18],...
'Callback',@(hObject,eventdata)abacus('RelativeHeight_Input_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'CreateFcn', {@local_CreateFcn, @(hObject,eventdata)abacus('RelativeHeight_Input_CreateFcn',...
                                    hObject,eventdata,guidata(hObject)), appdata} ,...
'Tag','RelativeHeight_Input',...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'DeleteEdgeMarkers_Check';

h46 = uicontrol(...
'Parent',h41,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Delete Edge Markers',...
'Style','checkbox',...
'Position',[30 61 131 23],...
'Callback',@(hObject,eventdata)abacus('DeleteEdgeMarkers_Check_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'ButtonDownFcn',blanks(0),...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} ,...
'DeleteFcn',blanks(0),...
'Tag','DeleteEdgeMarkers_Check',...
'KeyPressFcn',blanks(0),...
'FontSize',get(0,'defaultuicontrolFontSize'));

appdata = [];
appdata.lastValidTag = 'ExportMarkers_Button';

h47 = uicontrol(...
'Parent',h41,...
'FontUnits',get(0,'defaultuicontrolFontUnits'),...
'Units',get(0,'defaultuicontrolUnits'),...
'String','Export Markers',...
'Style',get(0,'defaultuicontrolStyle'),...
'Position',[228 12 96 30],...
'Callback',@(hObject,eventdata)abacus('ExportMarkers_Button_Callback',hObject,eventdata,guidata(hObject)),...
'Children',[],...
'Tag','ExportMarkers_Button',...
'CreateFcn', {@local_CreateFcn, blanks(0), appdata} );


hsingleton = h1;


% --- Set application data first then calling the CreateFcn.
function local_CreateFcn(hObject, eventdata, createfcn, appdata)

if ~isempty(appdata)
   names = fieldnames(appdata);
   for i=1:length(names)
       name = char(names(i));
       setappdata(hObject, name, getfield(appdata,name));
   end
end

if ~isempty(createfcn)
   if isa(createfcn,'function_handle')
       createfcn(hObject, eventdata);
   else
       eval(createfcn);
   end
end


% --- Handles default GUIDE GUI creation and callback dispatch
function varargout = gui_mainfcn(gui_State, varargin)

gui_StateFields =  {'gui_Name'
    'gui_Singleton'
    'gui_OpeningFcn'
    'gui_OutputFcn'
    'gui_LayoutFcn'
    'gui_Callback'};
gui_Mfile = '';
for i=1:length(gui_StateFields)
    if ~isfield(gui_State, gui_StateFields{i})
        error(message('MATLAB:guide:StateFieldNotFound', gui_StateFields{ i }, gui_Mfile));
    elseif isequal(gui_StateFields{i}, 'gui_Name')
```

```
        gui_Mfile = [gui_State.(gui_StateFields{i}), '.m'];
    end
end

numargin = length(varargin);

if numargin == 0
    % ABACUS
    % create the GUI only if we are not in the process of loading it
    % already
    gui_Create = true;
elseif local_isInvokeActiveXCallback(gui_State, varargin{:})
    % ABACUS(ACTIVEX,...)
    vin{1} = gui_State.gui_Name;
    vin{2} = [get(varargin{1}.Peer, 'Tag'), '_', varargin{end}];
    vin{3} = varargin{1};
    vin{4} = varargin{end-1};
    vin{5} = guidata(varargin{1}.Peer);
    feval(vin{:});
    return;
elseif local_isInvokeHGCallback(gui_State, varargin{:})
    % ABACUS('CALLBACK',hObject,eventData,handles,...)
    gui_Create = false;
else
    % ABACUS(...)
    % create the GUI and hand varargin to the openingfcn
    gui_Create = true;
end

if ~gui_Create
    % In design time, we need to mark all components possibly created in
    % the coming callback evaluation as non-serializable. This way, they
    % will not be brought into GUIDE and not be saved in the figure file
    % when running/saving the GUI from GUIDE.
    designEval = false;
    if (numargin>1 && ishghandle(varargin{2}))
        fig = varargin{2};
        while ~isempty(fig) && ~ishghandle(fig,'figure')
            fig = get(fig,'parent');
        end

        designEval = isappdata(0,'CreatingGUIDEFigure') || (isscalar(fig)&&isprop(fig,'GUIDEFigure'));
    end

    if designEval
        beforeChildren = findall(fig);
    end

    % evaluate the callback now
    varargin{1} = gui_State.gui_Callback;
    if nargout
        [varargout{1:nargout}] = feval(varargin{:});
    else
        feval(varargin{:});
    end

    % Set serializable of objects created in the above callback to off in
    % design time. Need to check whether figure handle is still valid in
    % case the figure is deleted during the callback dispatching.
    if designEval && ishghandle(fig)
        set(setdiff(findall(fig),beforeChildren), 'Serializable','off');
    end
else
    if gui_State.gui_Singleton
        gui_SingletonOpt = 'reuse';
    else
        gui_SingletonOpt = 'new';
    end

    % Check user passing 'visible' P/V pair first so that its value can be
    % used by oepnfig to prevent flickering
    gui_Visible = 'auto';
    gui_VisibleInput = '';
    for index=1:2:length(varargin)
        if length(varargin) == index || ~ischar(varargin{index})
            break;
        end

        % Recognize 'visible' P/V pair
        len1 = min(length('visible'),length(varargin{index}));
        len2 = min(length('off'),length(varargin{index+1}));
        if ischar(varargin{index+1}) && strncmpi(varargin{index},'visible',len1) && len2 > 1
            if strncmpi(varargin{index+1},'off',len2)
                gui_Visible = 'invisible';
                gui_VisibleInput = 'off';
            elseif strncmpi(varargin{index+1},'on',len2)
                gui_Visible = 'visible';
                gui_VisibleInput = 'on';
            end
        end
```

```
end

% Open fig file with stored settings.  Note: This executes all component
% specific CreateFunctions with an empty HANDLES structure.


% Do feval on layout code in m-file if it exists
gui_Exported = ~isempty(gui_State.gui_LayoutFcn);
% this application data is used to indicate the running mode of a GUIDE
% GUI to distinguish it from the design mode of the GUI in GUIDE. it is
% only used by actxproxy at this time.
setappdata(0,genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]),1);
if gui_Exported
    gui_hFigure = feval(gui_State.gui_LayoutFcn, gui_SingletonOpt);

    % make figure invisible here so that the visibility of figure is
    % consistent in OpeningFcn in the exported GUI case
    if isempty(gui_VisibleInput)
        gui_VisibleInput = get(gui_hFigure,'Visible');
    end
    set(gui_hFigure,'Visible','off')

    % openfig (called by local_openfig below) does this for guis without
    % the LayoutFcn. Be sure to do it here so guis show up on screen.
    movegui(gui_hFigure,'onscreen');
else
    gui_hFigure = local_openfig(gui_State.gui_Name, gui_SingletonOpt, gui_Visible);
    % If the figure has InGUIInitialization it was not completely created
    % on the last pass.  Delete this handle and try again.
    if isappdata(gui_hFigure, 'InGUIInitialization')
        delete(gui_hFigure);
        gui_hFigure = local_openfig(gui_State.gui_Name, gui_SingletonOpt, gui_Visible);
    end
end
if isappdata(0, genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]))
    rmappdata(0,genvarname(['OpenGuiWhenRunning_', gui_State.gui_Name]));
end

% Set flag to indicate starting GUI initialization
setappdata(gui_hFigure,'InGUIInitialization',1);

% Fetch GUIDE Application options
gui_Options = getappdata(gui_hFigure,'GUIDEOptions');
% Singleton setting in the GUI MATLAB code file takes priority if different
gui_Options.singleton = gui_State.gui_Singleton;

if ~isappdata(gui_hFigure,'GUIOnScreen')
    % Adjust background color
    if gui_Options.syscolorfig
        set(gui_hFigure,'Color', get(0,'DefaultUicontrolBackgroundColor'));
    end

    % Generate HANDLES structure and store with GUIDATA. If there is
    % user set GUI data already, keep that also.
    data = guidata(gui_hFigure);
    handles = guihandles(gui_hFigure);
    if ~isempty(handles)
        if isempty(data)
            data = handles;
        else
            names = fieldnames(handles);
            for k=1:length(names)
                data.(char(names(k)))=handles.(char(names(k)));
            end
        end
    end
    guidata(gui_hFigure, data);
end

% Apply input P/V pairs other than 'visible'
for index=1:2:length(varargin)
    if length(varargin) == index || ~ischar(varargin{index})
        break;
    end

    len1 = min(length('visible'),length(varargin{index}));
    if ~strncmpi(varargin{index},'visible',len1)
        try set(gui_hFigure, varargin{index}, varargin{index+1}), catch break, end
    end
end

% If handle visibility is set to 'callback', turn it on until finished
% with OpeningFcn
gui_HandleVisibility = get(gui_hFigure,'HandleVisibility');
if strcmp(gui_HandleVisibility, 'callback')
    set(gui_hFigure,'HandleVisibility', 'on');
end

feval(gui_State.gui_OpeningFcn, gui_hFigure, [], guidata(gui_hFigure), varargin{:});
```

```
    if isscalar(gui_hFigure) && ishghandle(gui_hFigure)
        % Handle the default callbacks of predefined toolbar tools in this
        % GUI, if any
        guidemfile('restoreToolbarToolPredefinedCallback',gui_hFigure);

        % Update handle visibility
        set(gui_hFigure,'HandleVisibility', gui_HandleVisibility);

        % Call openfig again to pick up the saved visibility or apply the
        % one passed in from the P/V pairs
        if ~gui_Exported
            gui_hFigure = local_openfig(gui_State.gui_Name, 'reuse',gui_Visible);
        elseif ~isempty(gui_VisibleInput)
            set(gui_hFigure,'Visible',gui_VisibleInput);
        end
        if strcmpi(get(gui_hFigure, 'Visible'), 'on')
            figure(gui_hFigure);

            if gui_Options.singleton
                setappdata(gui_hFigure,'GUIOnScreen', 1);
            end
        end

        % Done with GUI initialization
        if isappdata(gui_hFigure,'InGUIInitialization')
            rmappdata(gui_hFigure,'InGUIInitialization');
        end

        % If handle visibility is set to 'callback', turn it on until
        % finished with OutputFcn
        gui_HandleVisibility = get(gui_hFigure,'HandleVisibility');
        if strcmp(gui_HandleVisibility, 'callback')
            set(gui_hFigure,'HandleVisibility', 'on');
        end
        gui_Handles = guidata(gui_hFigure);
    else
        gui_Handles = [];
    end

    if nargout
        [varargout{1:nargout}] = feval(gui_State.gui_OutputFcn, gui_hFigure, [], gui_Handles);
    else
        feval(gui_State.gui_OutputFcn, gui_hFigure, [], gui_Handles);
    end

    if isscalar(gui_hFigure) && ishghandle(gui_hFigure)
        set(gui_hFigure,'HandleVisibility', gui_HandleVisibility);
    end
end

function gui_hFigure = local_openfig(name, singleton, visible)

% openfig with three arguments was new from R13. Try to call that first, if
% failed, try the old openfig.
if nargin('openfig') == 2
    % OPENFIG did not accept 3rd input argument until R13,
    % toggle default figure visible to prevent the figure
    % from showing up too soon.
    gui_OldDefaultVisible = get(0,'defaultFigureVisible');
    set(0,'defaultFigureVisible','off');
    gui_hFigure = matlab.hg.internal.openfigLegacy(name, singleton);
    set(0,'defaultFigureVisible',gui_OldDefaultVisible);
else
    % Call version of openfig that accepts 'auto' option"
    gui_hFigure = matlab.hg.internal.openfigLegacy(name, singleton, visible);
%     %workaround for CreateFcn not called to create ActiveX
%         peers=findobj(findall(allchild(gui_hFigure)),'type','uicontrol','style','text');
%         for i=1:length(peers)
%             if isappdata(peers(i),'Control')
%                 actxproxy(peers(i));
%             end
%         end
end

function result = local_isInvokeActiveXCallback(gui_State, varargin)

try
    result = ispc && iscom(varargin{1}) ...
             && isequal(varargin{1},gcbo);
catch
    result = false;
end

function result = local_isInvokeHGCallback(gui_State, varargin)

try
    fhandle = functions(gui_State.gui_Callback);
    result = ~isempty(findstr(gui_State.gui_Name,fhandle.file)) || ...
             (ischar(varargin{1}) ...
             && isequal(ishghandle(varargin{2}), 1) ...
```

```
            && (~isempty(strfind(varargin{1},[get(varargin{2}, 'Tag'), '_'])) || ...
               ~isempty(strfind(varargin{1}, '_CreateFcn'))) );
catch
    result = false;
end
```