

## SUPPLEMENTAL FIGURES

Wang, Ochoa, Khaliullin et al., Figure S1

	Promoter	Length	Reporter(s)	Expressing Tissue(s)	3' UTR	Allele(s)
Transgenes used in Germ Layer & Morphogenesis strains	<i>Pcnd-1</i> <sup>1</sup>	3230 bp	mCh::Plasma Membrane	Strong, specific expression in ~1/3 of neurons from embryonic stage	<i>unc-54</i>	<i>ltSi511</i>
	<i>Pdlg-1Δ7</i> <sup>2</sup>	3940 bp	DLG-1::GFP	Embryonic epidermis	<i>unc-54</i>	<i>ltSi249</i>
	<i>Pcnd-1</i>	3230 bp	mCh::histone (tandem with <i>Pdlg-1Δ7</i> mCh::histone)	Strong, specific expression in ~1/3 of neurons from embryonic stage	<i>unc-54</i>	<i>ltSi539</i>
	<i>Pdlg-1Δ7</i>	3940 bp	mCh::histone (tandem with <i>Pcnd-1</i> mCh::histone)	Embryonic epidermis	<i>unc-54</i>	<i>ltSi539</i>
	<i>Phlh-1</i> <sup>3</sup>	3345 bp	mCh::histone & GFP::histone (in tandem)	Body wall muscle	<i>tbb-2</i>	<i>ltSi507</i>
	<i>pha-4</i> fosmid <sup>4</sup>	33505 bp	PHA-4::GFP	Intestine and pharynx	<i>pha-4</i>	<i>stls10389</i>
Other transgenes generated for the project	<i>Pcnd-1</i>	3230 bp	GFP::histone (tandem with <i>Pdlg-1Δ7</i> GFP::histone)	Strong, specific expression in ~1/3 of neurons from embryonic stage	<i>unc-54</i>	<i>ltSi509</i>
	<i>Pdlg-1Δ7</i>	3940 bp	GFP::histone (tandem with <i>Pcnd-1</i> GFP::histone)	Embryonic epidermis	<i>unc-54</i>	<i>ltSi509</i>
	<i>Pelt-2</i>	2994 bp	mCh::histone & GFP::histone (in tandem)	Intestine	<i>tbb-2</i>	<i>ltSi506</i>
	<i>Phlh-1</i>	3345 bp	mCh::histone, GFP::histone (separate)	Body wall muscle	<i>unc-54</i>	<i>ltSi456, ltSi457</i>
	<i>Ppha-4+intron1</i>	6452 bp	mCh::histone (tandem with <i>Phlh-1</i> mCh::histone)	Weak pharynx, strong intestine	<i>tbb-2</i>	<i>ltSi504</i>
	<i>Ppha-4</i>	4157 bp	mCh::histone (tandem with <i>Phlh-1</i> mCh::histone)	Weak pharynx, variable intestine	<i>tbb-2</i>	<i>ltSi273</i>
	<i>Prgef-1</i>	4040 bp	mCh::Plasma Membrane	No embryonic expression Post-embryonic pan-neuronal	<i>unc-54</i>	<i>ltSi318</i>
	<i>Punc-33</i>	2003 bp	mCh::Plasma Membrane	All embryonic cells	<i>unc-54</i>	<i>ltSi316</i>
	<i>Punc-119+intron1</i>	4749 bp	mCh::Plasma Membrane	All embryonic cells	<i>unc-54</i>	<i>ltSi321</i>

**Footnotes:**

1. A HIS-24::mCherry reporter utilizing *Pcnd-1* (-3266 to -10 bp) is expressed in a subset of AB progeny cells starting at about the 100-cell stage. These cells include 97 neurons, 17 glia cells and 13 non-nervous system cells (for a map of *Pcnd-1* expression superimposed with the embryonic lineage see Murray and Bao et al., 2008). Note that the *C. elegans* embryo has 248 total neurons, so the *Pcnd-1* promoter directs expression in about 40% of neurons. These neurons include ventral cord motor neurons and some neurons in the nerve ring ganglion, and do not overlap with UNC-86-expressing neurons (egg-laying neurons, mechanosensory neurons, and chemosensory interneurons; Hallam and Singer et al., 2000).

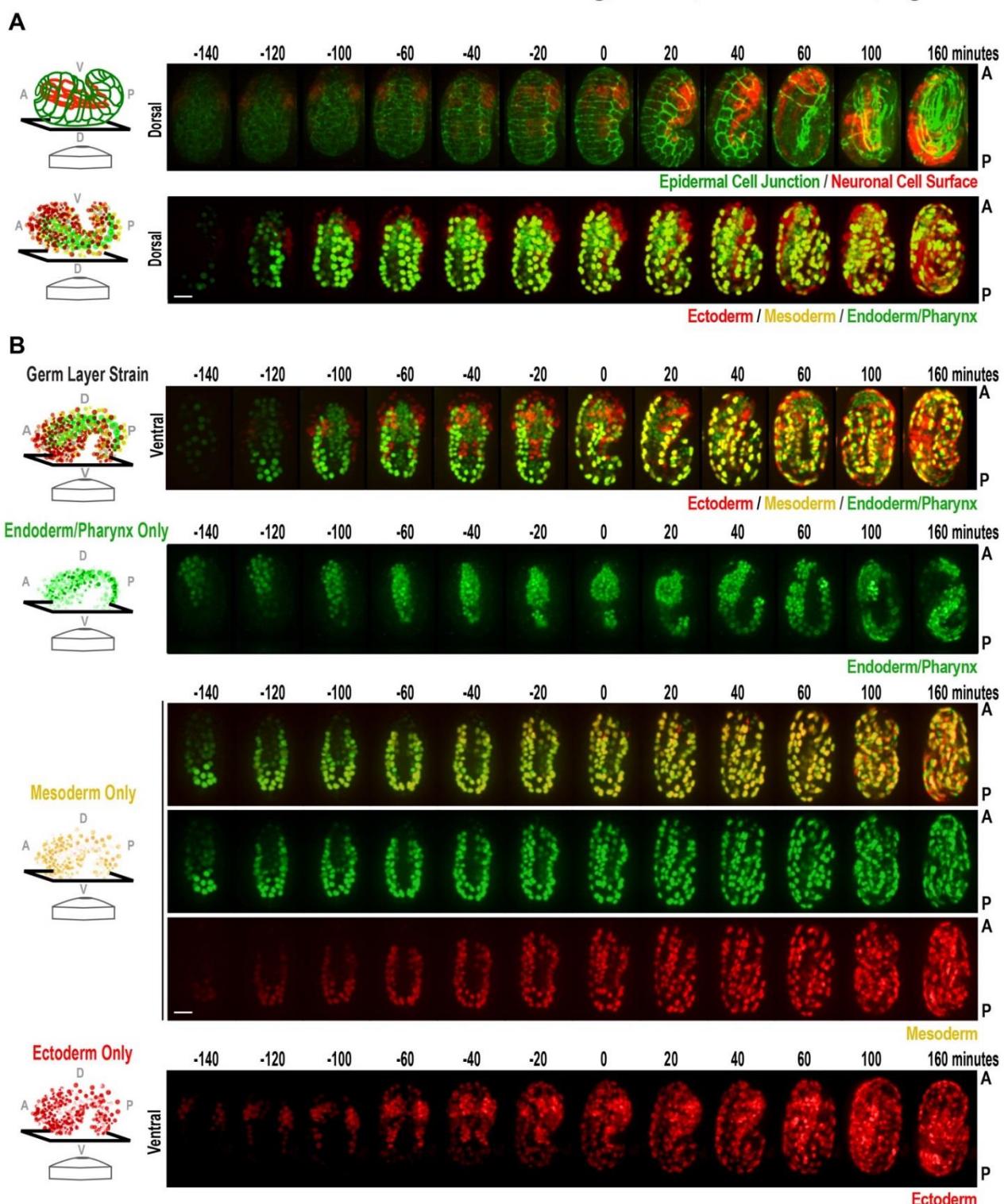
2. *Pdlg-1Δ7* drives expression in all epidermal cells (Sheffield et al., 2007)

3. A lacZ reporter utilizing *Phlh-1* (-3053 to + 70 bp) showed expression in all body wall muscle precursor cells with no ectopic expression (Krause et al., 1994). A HIS-24::mCherry reporter utilizing *Phlh-1* (-3053 to -4 bp) showed expression in all body wall muscle cells as well as some pharyngeal cells of the MS lineage (Murray and Bao et al., 2008; includes a map of expression superimposed with the embryonic lineage). In our case, *Phlh-1* (-3333 to +12 bp) reporter expression appears to be expressed in all body wall muscle precursors (Fig. S2B, bottom panels).

4. The fosmid recombineering transgene we used for PHA-4 is allele *stls10389* (Fakhouri et al. 2010, Zhong, Niu et al. 2010). It was made by tagging PHA-4 at the C-terminus with GFP, a Ty1 peptide and 3xFLAG in the fosmid WRM0617dE06. This fosmid is 33505 bp in length and contains the entire *pha-4* genomic cassette (Fakhouri et al. 2010; Zhong et al., 2010). Consistent with our own observations, this transgene likely recapitulates the endogenous *pha-4* expression pattern, which includes all pharyngeal, midgut and rectal precursors (Horner et al., 1998).

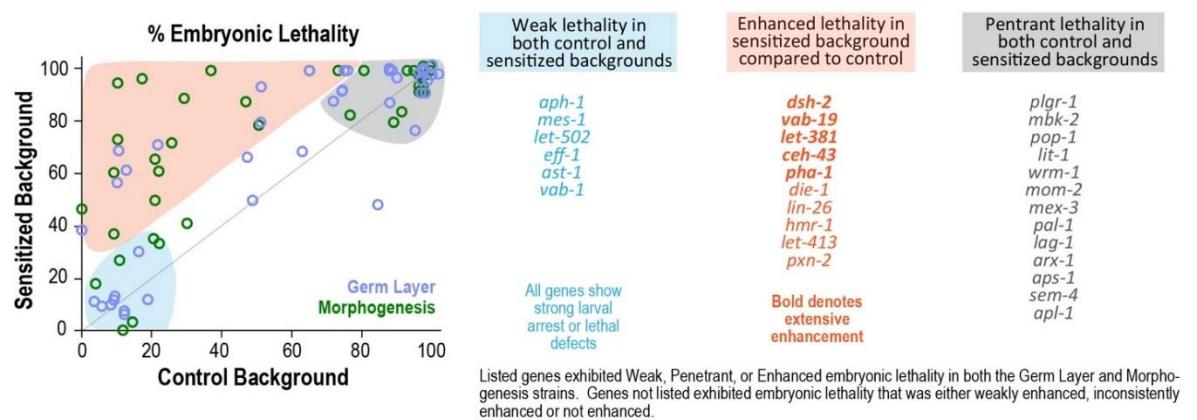
**Figure S1. List of transgenes constructed for this study.** Table listing the promoter/reporter combinations used in the transgenes constructed for this study, along with notes on the tissue expression of each marker. The combinations in the top half of the table (**green background**) were used in the transgenes in the Germ Layer and Morphogenesis strains. The combinations in the bottom half of the table (**gray background**) were tried and are available but were not used in our final strains.

Wang, Ochoa, Khaliullin et al., Figure S2



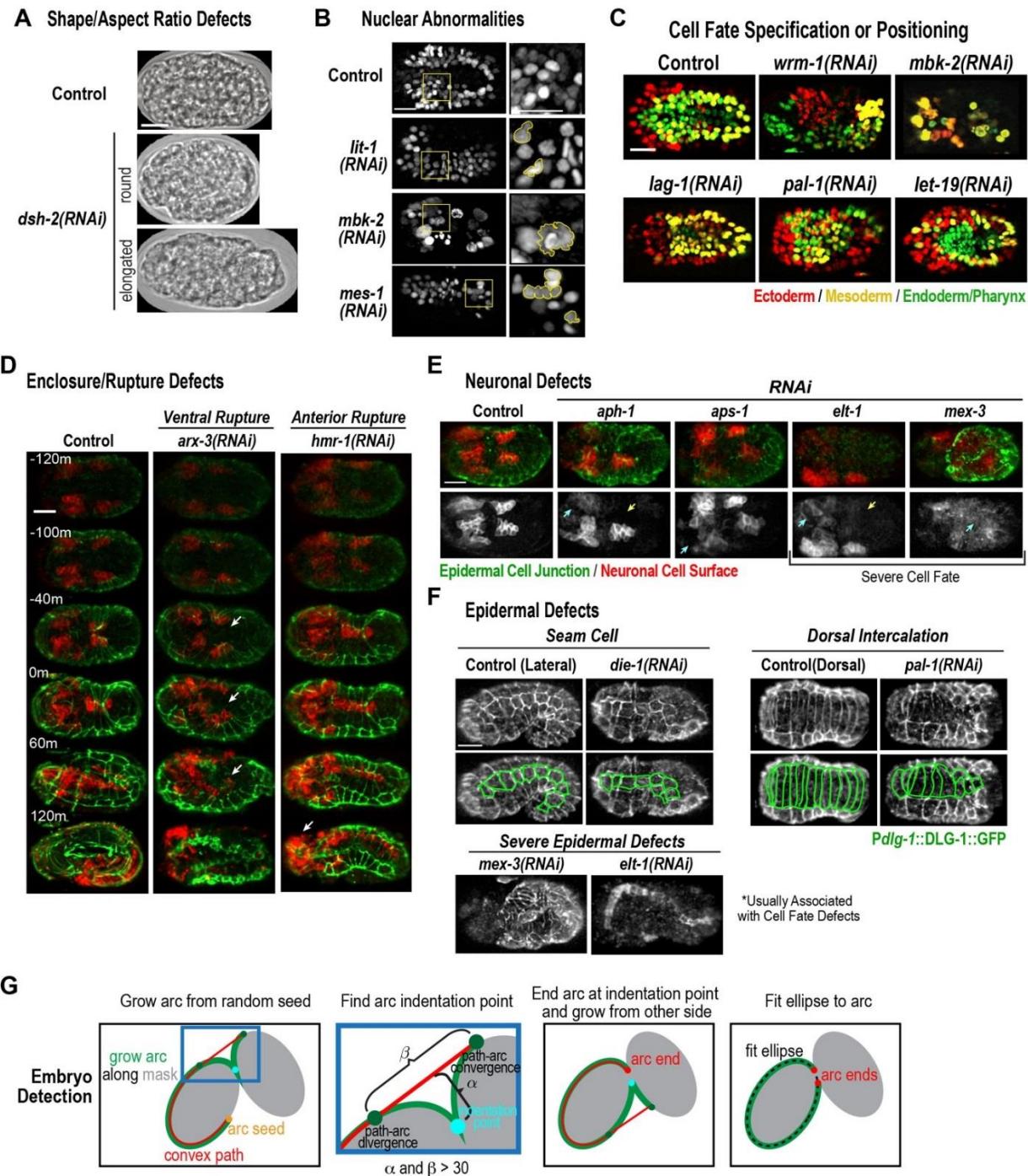
**Figure S2. Images showing development from a dorsal view in the Germ Layer and Morphogenesis strains and in embryos individually expressing the three transgenes that make up the Germ Layer strain.** (A) Maximum intensity projections showing a dorsal view (see schematics on left) of embryonic development in the Morphogenesis (**top**) and Germ Layer (**bottom**) strains acquired at the indicated timepoints relative to the comma stage (t=0 minutes). (B) Maximum intensity projections showing a ventral view (see schematics on left) of embryos from the Germ Layer strain (top panels; reproduced from Figure 2A for comparison) and strains individually expressing the three component transgenes that make up the Germ Layer strain (Ectoderm Only, strain OD1599; Endoderm/Pharynx Only, strain OD1598; and Mesoderm Only, strain OD1543). Scale bars are 10  $\mu$ m.

## Wang, Ochoa, Khaliullin et al., Figure S3



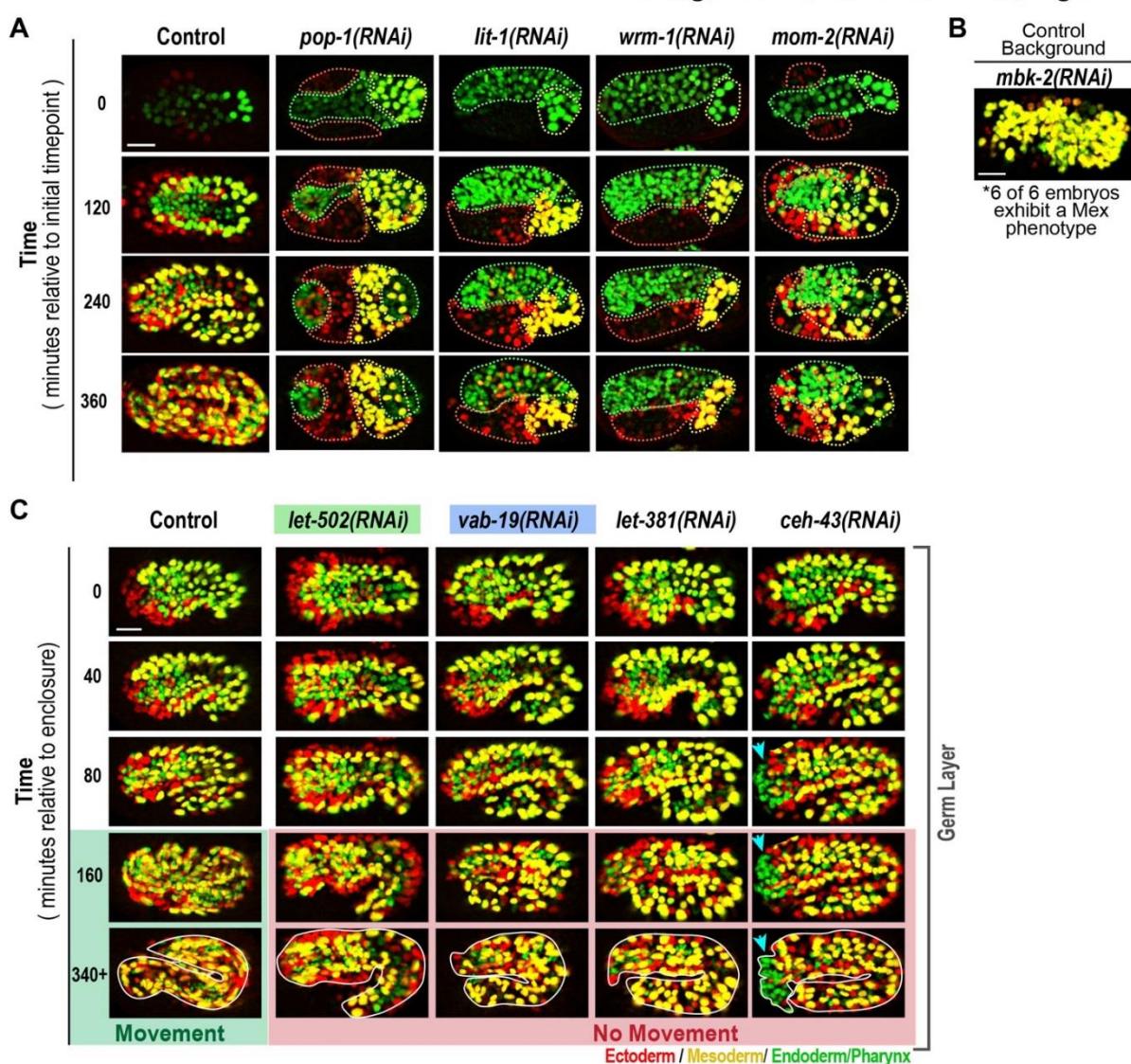
**Figure S3. Comparison of embryonic lethality in the Germ Layer and Morphogenesis strains for Control and Sensitized backgrounds. (A)** Graph (**left**) plots percent embryonic lethality in the control background (x-axis) versus percent embryonic lethality in the RNAi-sensitized background (y-axis) for each RNAi condition in the Germ Layer (**blue data points**) and Morphogenesis (**green data points**) strains. Shaded colored regions highlight the gene targets that exhibited weak embryonic lethality (<30%) in both backgrounds (**light blue**), penetrant embryonic lethality (>70%) in both backgrounds (**gray**), or showed enhanced embryonic lethality (20% or greater increase) in the sensitized background compared to the control background (**orange**). Genes in each of these color-coded cohorts are listed (**right**).

## Wang, Ochoa, Khaliullin et al., Figure S4



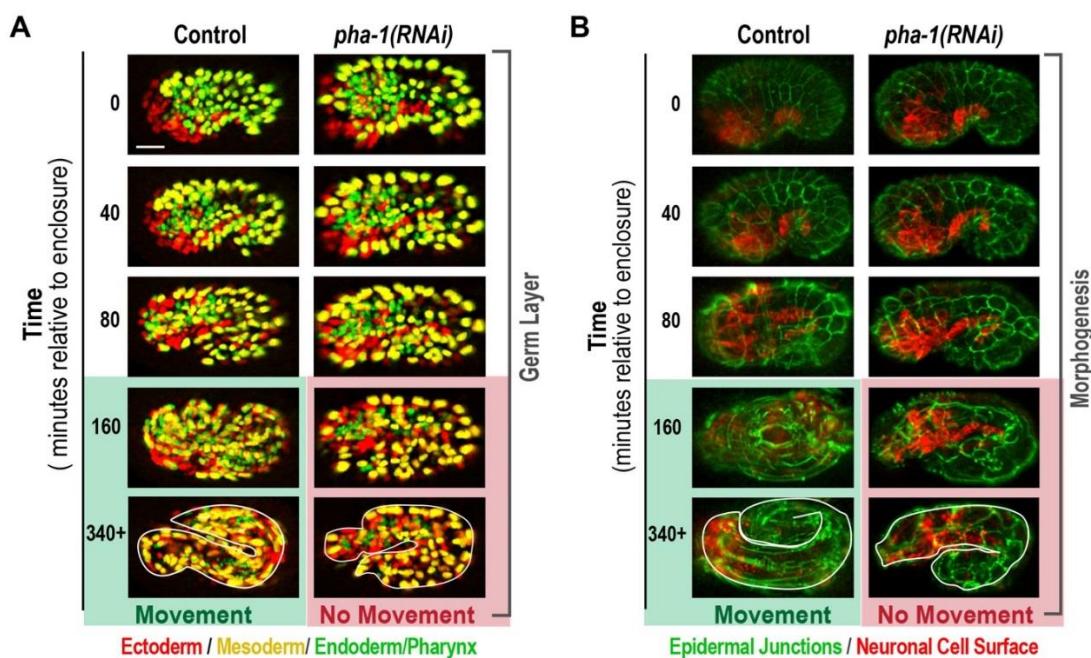
**Figure S4. Scoring of phenotypic features in the Germ Layer and Morphogenesis strains and automated cropping.** (A-F) Maximum intensity projections of fluorescence confocal images (B-F), or single plane brightfield images (A) of embryos from specific RNAi conditions scored with the indicated descriptors are shown to illustrate how each descriptor was scored. (A) Embryos were scored with a “shape/aspect ratio” defect if embryo shape deviated significantly different from controls. (B) Embryos (Germ Layer only) were scored with “nuclear abnormalities” if any nuclei were abnormal in shape, if micronuclei or lagging chromosomes were observed, or if nuclear marker intensity was uneven. (C) Germ Layer embryos were scored with a “cell fate specification or positioning” defect if the number or position of nuclei in the germ layers appeared to deviate from that in controls (based on visual assessment) prior to the onset of morphogenesis (comma stage). Control, *lag-1(RNAi)*, *pal-1(RNAi)* and *mbk-2(RNAi)* images are reproduced for comparison from Figure 5D-E. (D) Embryos were scored for a “ventral enclosure defect/ventral rupture” if the ventral epidermal/ectodermal cells moved towards the midline but failed to completely close the ventral opening, or an “anterior closure defect/anterior rupture” if the epidermal/ectoderm cells failed to move anteriorly to completely cover the head. (E) Embryos (Morphogenesis only) were scored with “neuronal defect” if neuronal organization was altered prior to elongation. (F) Embryos (Morphogenesis only) were scored with a “seam cell defect” if the position, structure or number of lateral seam cells was abnormal between the comma and 1.5-fold stages, with “dorsal intercalation defect” if the dorsal hypodermal cells (either one or two to all) failed to properly intercalate or to fuse after intercalation, and with “severe epidermal defect” if the pattern of cells that expressed the markers was severely disorganized. (G) In our automated cropping algorithm, a binary mask was generated from 8-bit brightfield images and individual embryos were detected and cropped out. Schematics detail the procedure used to iteratively detect embryos in the binary mask. Scale bar is 10 µm.

Wang, Ochoa, Khaliullin et al., Figure S5



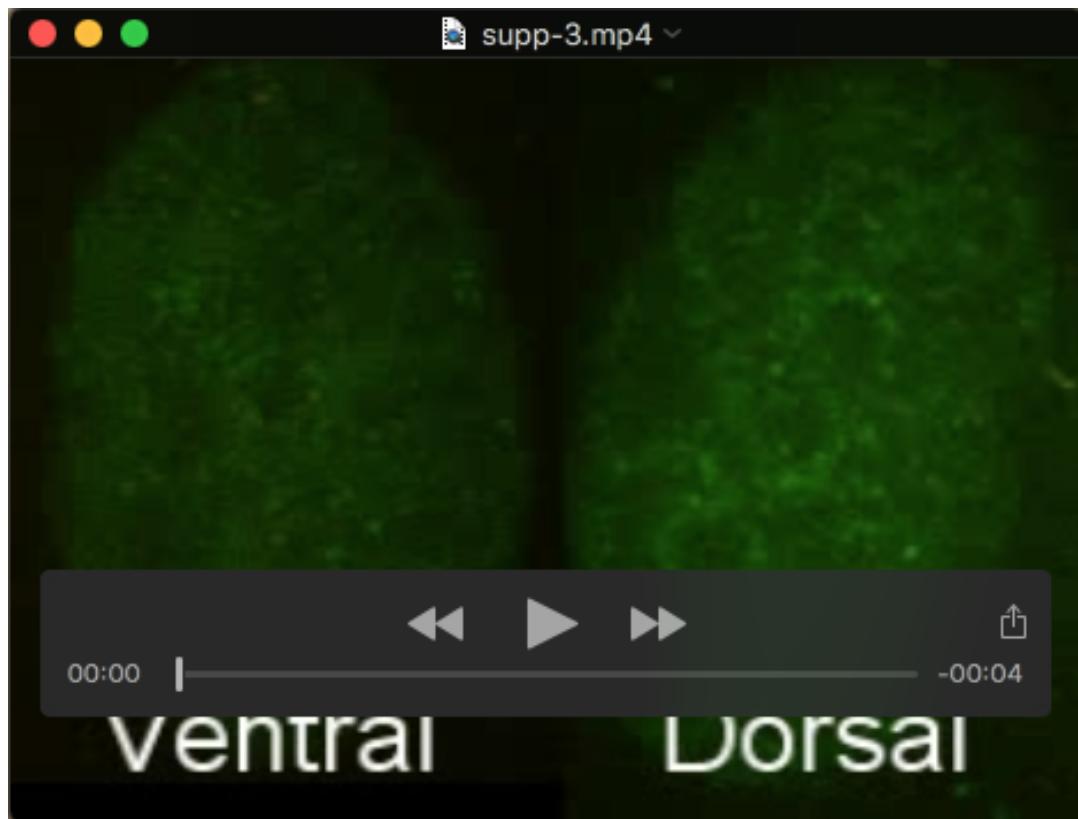
**Figure S5. Germ Layer strain phenotypes for WNT pathway components and proteins required for embryo elongation.** **(A)** Maximum intensity projections of confocal images of embryos from the Germ Layer strain with RNAi-sensitizing mutations after knockdown of genes encoding four components of the WNT signaling pathway. Dashed green, red and yellow outlines mark the locations where nuclei of the corresponding colors are observed. See text for phenotype descriptions. Final *wrm-1(RNAi)* and *lit-1(RNAi)* images are reproduced from Figure 5D. For the full sequences see movies available at Dryad repository: (<https://datadryad.org/review?doi=doi:10.5061/dryad.kb79121>). **(B)** Image shows the Germ Layer *mbk-2(RNAi)* phenotype in control background lacking the RNAi-sensitizing mutations. **(C)** Maximum intensity projections of confocal images of embryos from the Germ Layer strain after knockdown of four genes required for elongation past the 2-fold stage. Arrowheads in *ceh-43(RNAi)* point to the position where pharyngeal nuclei are leaking out of the head of the embryo. Elongating embryos were outlined in the final frame to make embryo shape at arrest more visible. In the *ceh-43(RNAi)* embryo, green pharyngeal nuclei are observed outside of the embryo. Green shading indicates successful elongation and movement of the embryo within the eggshell, whereas, red shading indicates arrest and failure to move. Scale bars are 10  $\mu\text{m}$ .

## Wang, Ochoa, Khaliullin et al., Figure S6



**Figure S6. Knockdown of *pha-1* leads to an elongation defect and a “pinched red head” phenotype.** (A, B) Maximum intensity projections of confocal images of embryos from the Germ Layer (A) and Morphogenesis (B) strains with RNAi-sensitizing mutations. Control embryo sequences are reproduced from Figure 6 and Figure S5 for comparison. PHA-1 is required for pharyngeal cells to attach to neighboring arcade cells to connect the pharynx to the buccal cavity, and *pha-1* mutants exhibit a penetrant Pun phenotype (Schnabel and Schnabel 1990, Fay 2004, Kuzmanov et al., 2014). Mosaic analysis suggested that PHA-1 may not be required in the pharynx but instead might promote pharyngeal development via a role in other cell types such as the epidermis (Kuzmanov et al., 2014). Interestingly, recent work has shown that, rather than being inherently required for development, PHA-1 is an antidote that counteracts the toxicity of a second protein called SUP-35 that is encoded with it on a selfish genetic element (Ben-David et al., 2017). Thus, *pha-1* knockdown phenotypes inform on the mechanism of action of the SUP-35 toxin. (A) In the Germ Layer strain, *pha-1* knockdown led to a “pinched red head” phenotype, possibly due to the defect in pharyngeal morphogenesis/attachment (**also see Movie S7**). (B) In the morphogenesis strain, PHA-1 inhibition led to a crumpled epidermis elongation defect, which our analysis suggests can arise as a consequence of a defect in mouth-pharynx attachment (*ceh-43*) or the failure to form transepithelial muscle attachments (*vab-19*). Green shading indicates successful elongation and movement of the embryo within the eggshell, whereas, red shading indicates arrest and failure to move. Scale bars are 10  $\mu$ m.

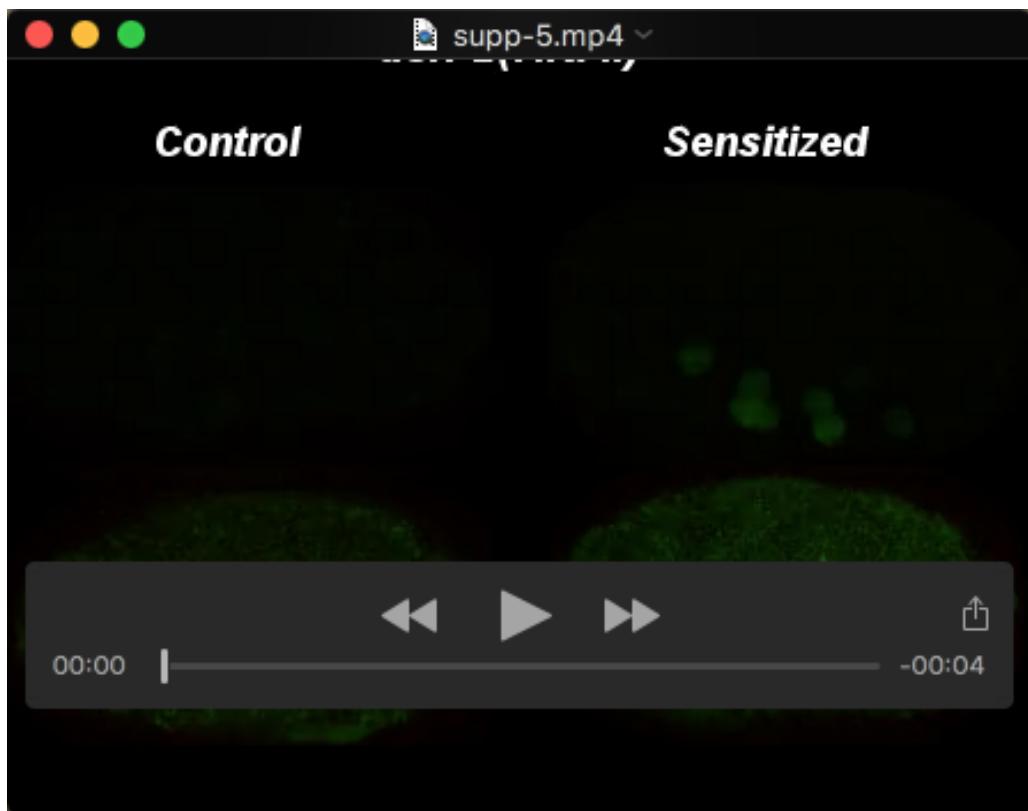
## Movies



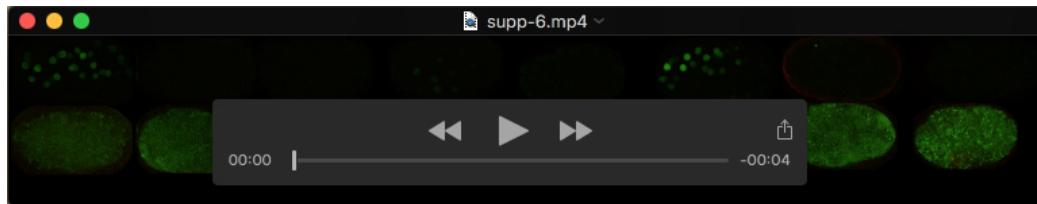
**Movie 1. Embryonic development imaged in the Morphogenesis strain.** Side-by-side ventral (**left**) and dorsal (**right**) views of development in control embryos from the Morphogenesis strain which has markers that localize to apical cell junctions in the epidermis (**green**) and the cell surface in about 1/3 of neurons (**red**). Images were acquired every 20 minutes. Playback is 7200X real time.



**Movie 2. Embryonic development imaged in the Germ Layer strain.** Side-by-side ventral (**left**) and dorsal (**right**) views of development in control embryos from the Germ Layer strain, which marks nuclei in the ectoderm (epidermis and ~1/3 of neurons; **red**), mesoderm (**yellow**) and endoderm/pharynx (**green**). Images were acquired every 20 minutes. Playback is 7200X real time.

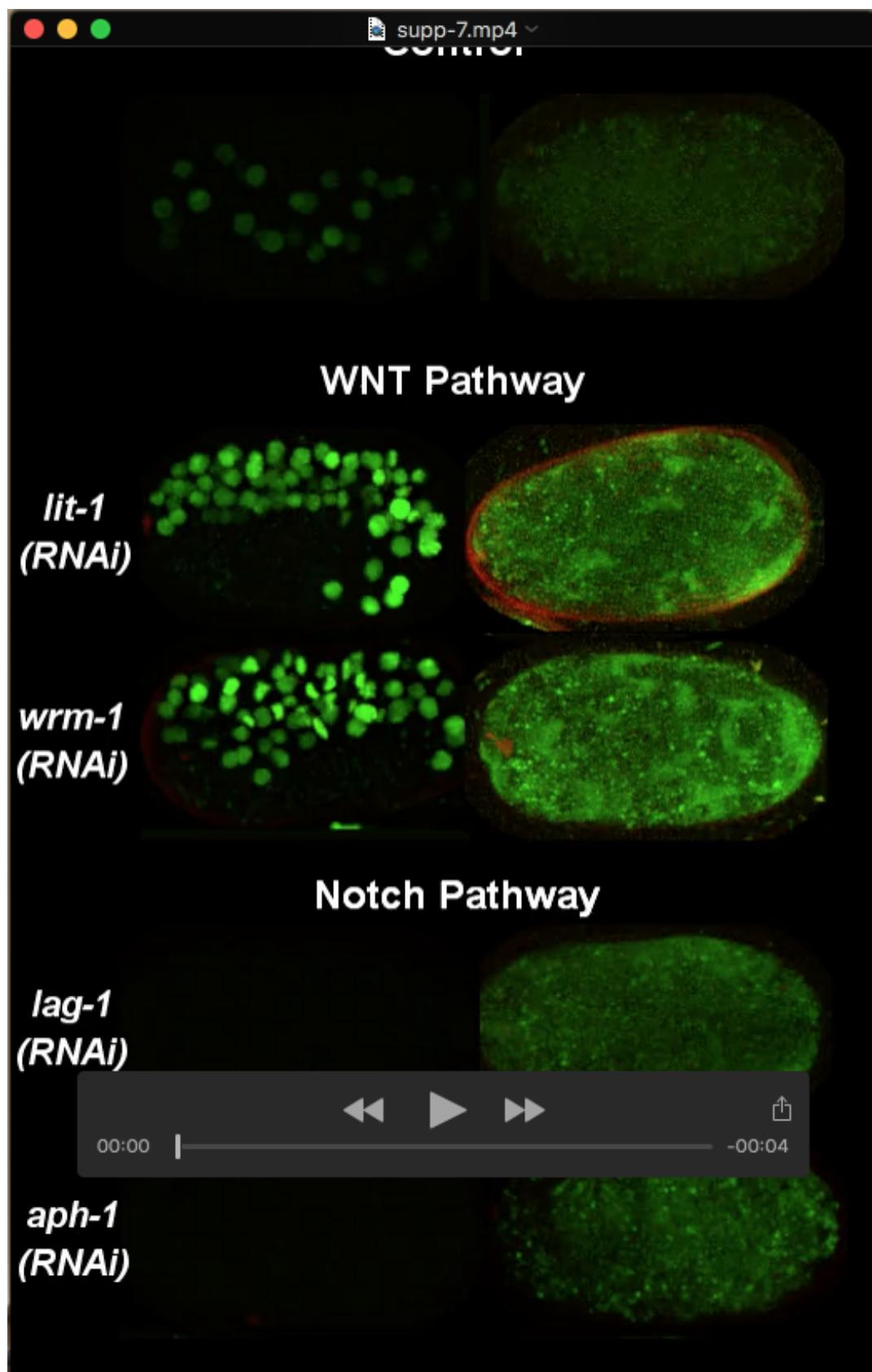


**Movie 3. Example of phenotypic enhancement in the RNAi-sensitized versions of the Germ Layer and Morphogenesis strains.** Side-by-side views showing the consequences of *dsh-2(RNAi)* in control (**left**) and RNAi-sensitized (**right**) versions of the Germ Layer (**top**) and Morphogenesis (**bottom**) strains. Images were acquired every 20 minutes. Playback is 7200X real time.



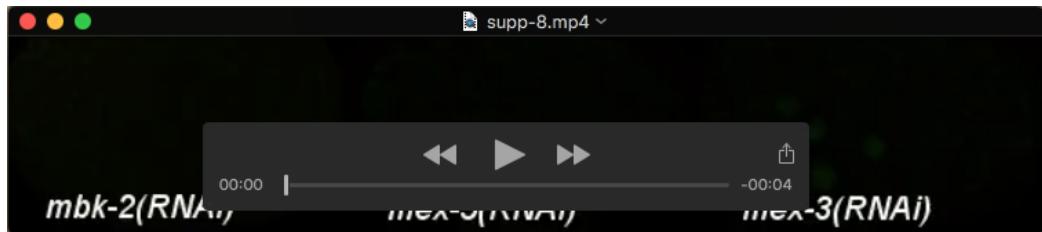
**Movie 4. Knockdown of genes required for cell fate specification and/or positioning leads to diverse, distinct phenotypes in the Germ Layer and Morphogenesis strains.**

Genes scored as having a defect in cell fate and/or positioning were partitioned into three classes based on analysis of nuclear counts: (1) genes whose inhibition leads to incomplete cell fate specification due to arrest prior to the completion of cell division, (2) genes whose inhibition leads to an abnormal nuclear pattern despite normal numbers of each type of nuclei and (3) bona fide genes cell fate specification genes. This movie shows side-by-side views of the phenotypes in the first two classes in the Germ Layer (**top**) and Morphogenesis (**bottom**) strains. See text for description of the *plrg-1*, *arx-1*, *arx-3*, *gex-2* and *die-1* knockdown phenotypes. APS-1, is an adaptin complex subunit involved in the formation of intracellular transport vesicles (Boehm and Bonifacino, 2001); *aps-1(RNAi)* embryos typically arrested without rupture, but with severe defects in epidermal and nervous system morphology. LET-19 is a component of the mediator complex previously shown to modulate the transcription of several genes involved in development (Wang et al., 2004; Yoda et al., 2005); *let-19* knockdown led to defects in epidermal morphology that were frequently accompanied by rupture of the epidermis at the embryo anterior. Playback is 7200X real time.



**Movie 5. Knocking down genes encoding WNT and NOTCH pathway components leads to distinct signature phenotypes in the Germ Layer and Morphogenesis strains.**

Movie presents side-by-side views of development in the Germ Layer (**left**) and Morphogenesis (**right**) strains for a control embryo (**top**) along with embryos in which genes encoding components of the WNT (*lit-1*, *wrm-1*) or NOTCH (*lag-1*, *aph-1*) signaling pathways have been knocked down. Images were acquired every 20 minutes. Playback is 7200X real time.



**Movie 6.** A Mex phenotype, similar to that in *mex-5/6* and *mex-3(RNAi)* embryos can be observed in *mbk-2* knockdown embryos. Movie presents side-by-side views of embryonic development in the Germ Layer strain after RNAi targeting *mbk-2* (**left**), *mex-5/mex-6* (**middle**) and *mex-3* (**right**). Images were acquired every 20 minutes. Playback is 7200X real time.



**Movie 7.** Filming in the Germ Layer and Morphogenesis strains enables functional classification of genes required for elongation. Movie presents side-by-side views of development in the Morphogenesis (**top**) and Germ Layer (**bottom**) strains for a control embryo (**left**) and embryos in which five genes required for elongation have been knocked down. Images were acquired every 20 minutes. Playback is 7200X real time.

**SUPPLEMENTAL TABLES****Table S1: Strains generated for this study****Table S1a. Available transgenic strains used in this study**

Strain Name	Transgene(s)	Genotype
<b>Composite Strains</b>		
OD1689 (Morphogenesis Reporter Strain, control background)	GFP-tagged cell junctions (epidermis); mCherry-labeled cell surface (1/3 of neurons)	<i>ltSi249[pOD1274/pSW098; Pdlg-1delta7::dlg-1-GFP::unc-54-3'UTR; cb-unc-119(+)]I; ltSi511[pOD2983/pSW207; Pcnnd-1::mCherry-PH::unc-54_3'UTR; cb-unc-119(+)]II</i>
OD2416 (Morphogenesis Reporter Strain, RNAi-sensitized background)	GFP-tagged cell junctions (epidermis); mCherry-labeled cell surface (1/3 of neurons); with RNAi-sensitizing mutations <i>nre-1(hd20)</i> & <i>lin-15b(hd126)</i>	<i>ltSi249[pOD1274/pSW098; Pdlg-1delta7::dlg-1-GFP::unc-54-3'UTR; cb-unc-119(+)]I; ltSi511[pOD2983/pSW207; Pcnnd-1::mCherry-PH::unc-54_3'UTR; cb-unc-119(+)]II; nre-1(hd20)X; lin-15b(hd126)X</i>
OD1719 (Germ Layer Reporter Strain, control background)	mCherry-tagged histone (epidermis and 1/3 of neurons); yellow (GFP & mCherry) histone (muscle); GFP::PHA-4 (pharynx and intestine)	<i>stls10389 [pha-4::TGF(3E3)::GFP::TY1::3xFLAG inserted into fosmid WRM0617dE06 as C-terminal protein fusion]; ltSi539[pOD1519/pSW224; Pdlg-1Δ7::mCherry::his-72::unc-54_3'UTR; Pcnnd-1::mCherry::his-72::unc-54_3'UTR; cb-unc-119(+)]II; ltSi507[pOD1492/pSW201; Phlh-1::GFP::his-72::tbb-2_3'UTR, Phlh-1::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]IV</i>
OD1854 (Germ Layer Reporter Strain, RNAi-sensitized background)	mCherry-tagged histone (epidermis and 1/3 of neurons); yellow (GFP & mCherry) histone (muscle); GFP::PHA-4 (pharynx and intestine); with RNAi-sensitizing mutations <i>nre-</i>	<i>stls10389 [pha-4::TGF(3E3)::GFP::TY1::3xFLAG inserted into fosmid WRM0617dE06 as C-terminal protein fusion]; ltSi539[pOD1519/pSW224; Pdlg-1Δ7::mCherry::his-72::unc-54_3'UTR; Pcnnd-1::mCherry::his-72::unc-54_3'UTR; cb-unc-119(+)]II; ltSi507[pOD1492/pSW201; Phlh-1::GFP::his-72::tbb-2_3'UTR, Phlh-1::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]IV; nre-1(hd20)X; lin-15b(hd126)X</i>

---

*1(hd20) & lin-  
15b(hd126)*

---

**Intermediate Strains**

OD1593	mCherry-labeled cell surface (1/3 of neurons)	<i>ltSi511[pOD2983/pSW207; PcnD-1::mCherry-PH::unc-54_3'UTR; cb-unc-119(+)]III</i>	6x
OD1591	GFP-tagged cell junctions (epidermis)- also in pharynx and intestine at post-embryonic stage.	<i>ltSi249[pOD1274/pSW098; Pdlg-1delta7::dlg-1-GFP::unc-54_3'UTR; cb-unc-119(+)]I</i>	8x
OD1599	mCherry-tagged histone (epidermis and 1/3 of neurons)	<i>ltSi539[pOD1519/pSW224; Pdlg-1Δ7::mCherry::his-72::unc-54_3'UTR; PcnD-1::mCherry::his-72::unc-54_3'UTR; cb-unc-119(+)]III</i>	6x
OD1592	yellow (GFP & mCherry) histone (muscle)	<i>unc-119(ed3)III; ltSi507[pOD1492/pSW201; Phlh-1::GFP::his-72::tbb-2_3'UTR, Phlh-1::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]IV</i>	6x
OD1598	GFP::PHA-4 (intestine & pharynx); previously generated insertion (Fakhouri et al., 2010 (PMID: 20714352)) that we backcrossed 6X	<i>stls10389 [pha-4::TGF(3E3)::GFP::TY1::3xFL AG inserted into fosmid WRM0617dE06 as C-terminal protein fusion]</i>	6x

---

**Table S1b. Available transgenic strains made for this study, but not used**

Strain Name	Expressing Transgene(s)	Genotype	Out-crossed
<b>Nuclear reporters</b>			

OD1557	mCherry-tagged histone (post-embryonic pan-neuronal)	<i>ltSi517[pOD1489/pSW19 8; Prgef-1::mCherry::his-72::unc-54_3'UTR; cb-unc-119(+)]l; unc-119(ed3)]III</i>	0x
OD1433	mCherry-tagged histone (body wall muscle)	<i>ltSi456[pOD1486/pSW19 5; Phlh-1::mCherry::his-72::unc-54_3'UTR; cb-unc-119(+)]l; unc-119(ed3)]III</i>	0x
OD1434	GFP-tagged histone (body wall muscle)	<i>ltSi457[pOD1487/pSW19 6; Phlh-1::GFP::his-72::unc-54_3'UTR; cb-unc-119(+)]l; unc-119(ed3)]III</i>	0x
OD1596	yellow (mCherry and GFP) histone (intestine)	<i>ltSi506[pOD1491/pSW20 0; Pelt-2::GFP::his-72::tbb-2_3'UTR, Pelt-2::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]IV</i>	6x
OD1595	GFP-tagged histone (epidermis and 1/3 of neurons)	<i>ltSi509[pOD2982/pSW20 3; Pdlg-1Δ7::GFP::his-72::unc-54_3'UTR; Pcdn-1::GFP::his-72::unc-54_3'UTR; cb-unc-119(+)]III</i>	6x
OD1007	mCherry-tagged histone (body wall muscle, pharynx, and intestine -- note pharynx expression is weak, and intestine is variable among lines; this line has strong intestine expression)	<i>ltSi273[pOD1272/pSW09 4; Phlh-1::mCherry::his-72::unc-54_3'UTR; Pphalpha-4::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]l; unc-119(ed3)]III</i>	0x
OD1540	mCherry-tagged histone (body wall muscle, pharynx, and intestine -- note pharynx expression is weak)	<i>ltSi504[pOD1294/pSW10 8; Phlh-1::mCherry::his-72::unc-54_3'UTR; Pphalpha-4intron1::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]l; unc-119(ed3)]III</i>	0x

---

**Cell morphology reporters**

---

OD1085	mCherry-labeled cell surface (embryonic, all cells; post-embryonic, appears to be pan-neuronal)	<i>ltSi316[pOD1293/pSW10 1s; Punc-33::mCherry-PH(PLCdelta1)::unc-54_3'UTR; cb-unc-119(+)]III; unc-119(ed3)]III</i>	0x
OD1087	mCherry-labeled cell surface (embryonic, no expression; post-embryonic, pan-neuronal)	<i>ltSi318[pOD1296/pSW12 6; Prgef-1::mCherry-PH(PLCdelta1)::unc-54_3'UTR; cb-unc-119(+)]III; unc-119(ed3)]III</i>	0x
OD1106	mCherry-labeled cell surface (embryonic, all cells; post-embryonic, appears to be pan-neuronal)	<i>ltSi321[pOD1295/pSW12 5; Punc-119::mCherry-PH(PLCdelta1)::unc-54_3'UTR; cb-unc-119(+)]III unc-119(ed3)]III</i>	0x
<b>Composite reporters</b>			
OD1639	GFP-tagged cell junctions (epidermis); mCherry-labeled cell surface (1/3 of neurons); yellow (GFP & mCherry) histone (muscle)	<i>ltSi249[pOD1274/pSW09 8; Pdlg-1delta7::dlg-1-GFP::unc-54_3'UTR; cb-unc-119(+)]I;</i> <i>ltSi511[pOD2983/pSW20 7; PcnD-1::mCherry-PH::unc-54_3'UTR; cb-unc-119(+)]III;</i> <i>ltSi507[pOD1492/pSW20 1; Phlh-1::GFP::his-72::tbb-2_3'UTR, Phlh-1::mCherry::his-72::tbb-2_3'UTR; cb-unc-119(+)]IV</i>	6x

**Table S2: Summary of phenotypic data for the 40-gene test set.** This document contains an index and 7 additional tabs. **S2.1-Embryonic Lethality:** Summarizes embryonic lethality and larval defects for each RNAi condition in the 4 strain backgrounds. **S2.2-Defects Scored:** provides a description of each phenotypic defect and how it was scored. **S2.3-Phenotype Scoring:** Includes quantification of all manually scored phenotypic characteristics for each gene in the 4 tested strain backgrounds (Germ Layer strain with RNAi-sensitizing mutations, Morphogenesis strain with RNAi-sensitizing mutations, Germ Layer strain control background, and Morphogenesis strain control background). **S2.4-Expected-Observed Summary:** Provides a comparison of the embryonic lethality and phenotypic defects observed in this study with annotated embryonic lethality and observed phenotypes by RNAi or mutant allele previously cataloged on WormBase. **S2.5-Gene Expression-Phenotypic Group:** shows the previously described temporal gene expression profile plots (Levin et al. 2012) for the genes in the broad phenotypic groups defined by the phenotypic scoring in this study. **S2.6-Raw Embryonic Lethality Data:** raw data counts of embryonic lethality, abnormal hatched L1, normal hatched L1 for the four considered strains. **S2.7-Oligo list:** sequences of the forward and reverse oligos used for the production of dsRNAs targeting each of the tested genes.

[Click here to Download Table S2](#)

## PROMOTER SEQUENCES

### Pcnd-1 (a subset of neurons, 3230 bp):

cagctatgacacgtggcttagtaatacttgcggcataacttgttgtgataactgttatccatccggctgtattttattcaga  
 agtcatgaaggaggcgatgaacacctgtggaaagagacaatataggatatctgtcaataaaaaaaaagatataatggtaacagg  
 atatgtgtatttaatgtattttacatgcgcgatataatgtgaatgttgcgcattttgttgcataataatttgtttatttgacatattatcaca  
 aaattggaaaagttcaggagagaaacacttcacaaaaaaattttcggaaatacccttatttgaaaaggtttgcggacaattcgt  
 caaaaaattaaactatggcaatgtgcaaaaaaaaataatttgaattttccaaaactatttgttagatctagaaaaaaatccctttaacg  
 atattcccttcattaattaatttgaattttataaagtgtggaaaatattttactatatcttctatatacccaattttgtacc  
 aaatcatcacctaggcaccatattgttatttcattataaaccgcaccgcgttgccttcctccaccgatcggcttgcggatt  
 aattacagacttgcattcatcataaaaaataccgcggctactccgtaaaacgttcaacacatacggtccaaatccgtttagaaatgt  
 tgacagtgacaaggaatttgtctaatttgtgtatgcattgcgttgcggtaaccaactgtcaaggatggaattcaacttagggatt  
 atgagtcaagaggtgtggaaaatttttgttatttgtttagcagaacaaaaggatataatgtgactcaaattgtgaggaaatgtgataatca  
 atttgtgaaatttttgtttagaataatccatgaaacttaattttcgttgcggatggatggatggacttgcggatggatggatgg  
 gtggatttttgttgcggatggatggatggatggatggatggatggatggatggatggatggatggatggatggatggatggatgg  
 cgggccactatttaagtgttctgtgttttccatgtctcaaaaagatgaaatggatggacttgcggatggatggatggatggatgg  
 catggatggatggactaagacatttgcggacttgcggatggatggatggatggatggatggatggatggatggatggatggatgg  
 ggaagaagggtgtttagaaaaaaaatagagggaaagaaaaaaaacagttatgttataatggaaaatgttataatggatggatgg  
 gagattcttcattactcattttgcgcgaaagtggatggatggatggatggatggatggatggatggatggatggatggatggatgg  
 gctgaagttttccaaaacttagtcttgcggatggatggatggatggatggatggatggatggatggatggatggatggatggatgg  
 accaacccttcattatgtttaaaacaaaatgtgactaataaagtgtctaaaaggctactgttgcggatggatggatggatggatgg  
 gcagattgtgaaatgtctcaaaaacatgcggatggatggatggatggatggatggatggatggatggatggatggatggatggatgg  
 ccaatcagttccgcgtcaacaaatctgtgtatcttgcggatggatggatggatggatggatggatggatggatggatggatggatgg

ttacagaatgtgtacccttataaaaacccttctggtaattagatgatatctgcattactgatgcacatcattgcataattggagaaat  
atgcataacaactttactggatgtactactctattcgataacggcgtatcgatagatcctatagaacaaaattcagactata  
tgttttcactctggaccctcactctcggaaatcgaaaaacgatgatattgaaatggaaagaaataagacaatctatgatctaatttc  
aaatcagaatttgatttcagaatttttggaaatattcatattgatgtattgttattgaccctgcaaaagtctcacagatacaatg  
agatcccagaagaacactggattatgtcaggaaggaagaaatattacagttgacacccctggatgttaggattacagagattt  
cattgtttcatgaaacttgcacacctgctattgacattgtgattgaaaaggtaaatattgaaactgacaagttctagtaatggc  
actctacacccttggtaggcggaaattaatggcaattccagaaattcagggcaatttcaaatctatttatatttgcggaaattttaaattac  
aaattgcaattgcgtgcatatctgacatgctccggatctaacagaagatcaacgaatgagatagtctactgttttaggcaaata  
acatgtaatcggtgacacgcatgtttgggtgagataaaatcatactgtaattaatatggagtagtgtgaactaccgtggacgattc  
tagcatgttcaagattttaaattacaagcacgcacgcgtttgttaactggatcggttccagaacaaaattttatctattggagggatt  
tattgaaatgtggAACACCCATTCCACACATGTTATAAGCTCGTATTAAATTAGTCATAGTGTGATCATGTTACAAGTTGGC  
ACATACATTCAAAAATATTAAACTTTGGATGCAGTTTAGTGGACATAATTAAAGAGTCCTTGCCTGGTACTTAGAACATT  
TGATATTGTTCTAAAAAGTAGATAAAGTAGAGCTAAAGTTTTAAAAGTATTTCATGCAAGGGCTATTAAAGCAAGTTCT  
AAAATCTGTTGAAGACACATCAACATTAAACTTTGCAGGTTCATCTGTAATTATCTTTAGAACTTTAATGCTATCTGGAA  
ACCATTCCATCCTCCCAACGAACATCCCGCCCTTGTACACCACGGCTCTGTCACCCAGCTCCACCGCCCCGCCCTCC  
TTCTGTGAACCATCCACCGATCAGTCATTCTTCACTATACTTCCTGTCATCCAGTTATTTCTACA

### Pdlg-1Δ7 (embryonic epidermis, 3940 bp):

ctcgccacaacttttggcgcattgtcccgcatgaaaaatggtacaatacgtacgcacgttcgtatcttccgcattgtcgactaat  
gtcccggtggctgtacatgtgaataattcac  
atgctggccgggtctgtccacttcatctgaccattatgttcgacactgtgattgttcaagtgactgttctgttttgcagtt  
ttggaaatcaagtggctattggaaatataatttttgtgaactcaaattgcagacgtgtgtgaccattaatttcaactatacaattttatg  
gcattatttctaataaaaaccccatttgttagataaaaattgttagtgttcaatgaaacatacatttagagcgaagataatttaacta  
tttaacttaaagagacgtatgggggtgttaacaatt  
aatatgttaatttaataaaaacttaatccaaaattttggatttttttcaggactcattatgaaacccctccagggagccaaaaccctccg  
aacttaatcccttatttaattcagaatttacaaaatccaaagataaaaatccataatagttactaattttgaatttagatga  
caacccctcaaaaacaatcgactccccgtccctaccaggccctccgatggcgggttgcgcctcgctataataaaaacc  
accggcggagggtgggtgtccaataaaacgttgaagacccccccaccgttatccaaacgccttcccttacccatcttctactca  
agcatatctcagatcttcacacacacagttccctctgttagataactatcttagatgtcgacccgccccgaggaaagac  
gagaggaatattatattttcccccattccaaacctactgaatcataccaaatgttggggatgtccgaacaaaaataaacat  
atccatgactcgacac

Pelt-2 (intestine, 2994 bp):

agtttaatttaaggcagtatttaagaaaatacactctcgaaagcatttgaattttgaattttaaactgctgatgttttagtgccactgtttcatgtataatgttgtataatgctgatgtttctggcattctaataaaaatagaactagaaaaatagattataga

### **Phlh-1 (body wall muscle, 3345 bp):**

Ppha-4 (intestine and weak pharynx, 4157 bp):

ggcccaaatttatgaccaaatgaaaatgaaaatttgaacacgactggtgtgaagcttaaaaaatagataagatttaact  
agtactgcaatgcgaatttgagcagtctttcaatttagcatatccaaaacagttctaggtgaatgaaacccctgctgatgt  
cqatatqtaqqaaaatqcaqcttcttqatccacctqcttcattqcattctatctcatacaqtaqcqcqcqcqtcagtqqqc

### Ppha-4 + intron1 (pharynx and intestine, 6452 bp):

ggcccaaatttatgaccaaataaaaatgtttgaacacgcactggtggtgaagcttaagaaaaatacgataagatttact  
agtactgcaatgcgaatttgagcagttcttctcaatttagcatatccaaaacagttcttaggtgaatgaaaccctgctgatgt  
cgatatgttaggaaaatgcagcttcttgcattccacctgcttcatttgcatatctcatacagtgacgcgcgcgcgtcgtggc  
qcaccgtattctcccttccctccqagaqqaqqacqtqcttagatcctttqacatttcttcgcgtcttttgtgaccaaaaqtatt



### Prgef-1 (post-embryonic pan-neuronal):

### Punc-119 + intron1 (embryonic all cells, 4749 bp)

### Punc-33 (embryonic all cells, 2003 bp):

## CROPPING PROGRAM INSTRUCTIONS

We provide two repositories that contain software:

### 1. Zenodo repository (<https://zenodo.org/record/1475442#.W9jvApNKiUJ>), which contains:

- **embryoCropUI\_WINDOWS**- *PC compatible executable version of graphical user interface (GUI)* version of our software, which can accommodate *individual image stacks from a range of imaging platforms*. This crops, rotates anterior-posterior, processes for drift correction, background subtraction and attenuation correction for individual image stacks. Cropped images are automatically saved to a “crop” file in the location of the file selected to be cropped.
- **embryoCropUI\_MacOS**- *Mac compatible executable version of graphical user interface (GUI)* version of our software, which can accommodate *individual image stacks from a range of imaging platforms*. This crops, rotates anterior-posterior, processes for drift correction, background subtraction and attenuation correction for individual image stacks. Cropped images are automatically saved to a “crop” file in the location of the file selected to be cropped. *(requires MacOS X10.11)*
- **Test\_files**- Contains two test files: a multi-tif format file and a folder with a tif image series, which can be used to test the GUI on your system (details for testing can be found in the instructions)
- **OpenAndCombine\_embs.ijm**- FIJI script that enables assembly of maximum intensity projections for multiple embryos within one viewer. Requires **ImageJ**. *Note this works with our file structure and will need to be modified to accommodate your file structure (see instructions).*
- **Instructions**- Instructions for installation and running GUI (embryoCropUI) and FIJI script (OpenAndCombine.ijm) (this file)

### 2. Github repository ([https://github.com/renatkh/embryo\\_crop.git](https://github.com/renatkh/embryo_crop.git)), which contains source code for embryoCropUI (embryoCropUI.py) and screenCrop.py:

- **embryoCropUI.py**: This is the source code for the user-friendly executable version described above, which crops, rotates anterior-posterior, processes for drift correction, background subtraction and attenuation correction for individual image stacks.
- **screenCrop.py**: This batch version of the embryoCrop.py program takes in multiwell, multipoint imaging data and crops, rotates anterior-posterior, processes for drift correction, and performs background subtraction and attenuation correction in batch. Requires **Python3, virtual environment, and .csv** file with the file format and condition specifications delineated. Saves cropped files to new “Cropped” folder with file structure as specified in .csv file. *This program is specific to our file structure and acquisition parameters, but can be modified to accommodate similarly structured data if users have Python expertise- see instructions below for key variables.*
- **README file**- Instructions for cloning repository, setting up virtual environment and running embryoCrop.py., and configuring necessary requirements to run screenCrop.py.

**\*\*Instructions for all programs can be found within each repository and are reproduced below for completeness:**

## Zenodo repository

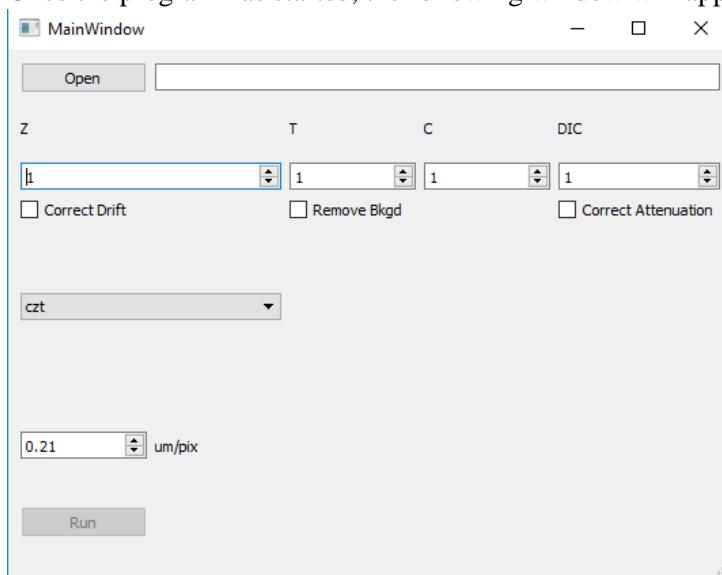
(<https://zenodo.org/record/1475442#.W9jvApNKiUI>)

- embryoCropUI\_WINDOWS
- embryoCropUI\_MacOS-requires MacOS X10.11
- Test\_files
- OpenAndCombine\_embs.ijm
- Instructions

## DETAILED INSTRUCTIONS

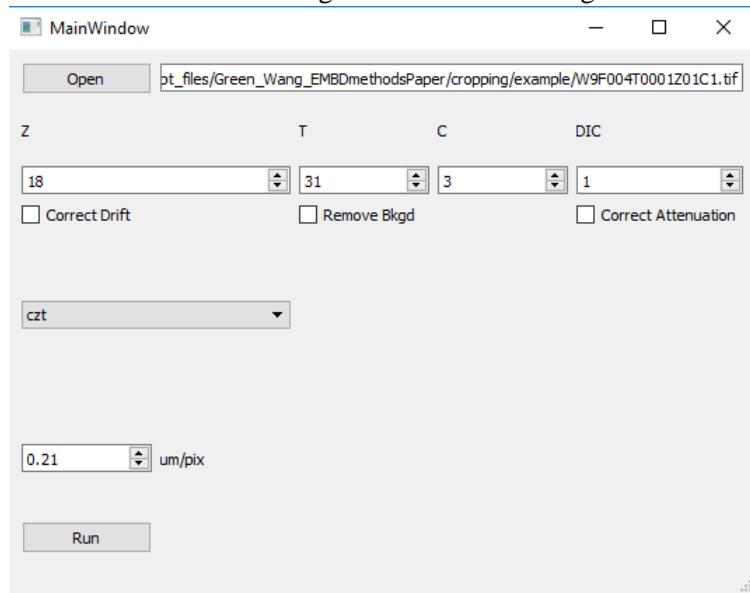
### 1. Running embryoCropUI.exe (MacOS and Windows)

1. Download Window or MacOS folder, unzip and navigate to find the embryoCropUI executable (...\\embryoCropUI \_WINDOWS\\embryoCropUI\\embryoCropUI.exe) or (...\\embryoCropUI \_MacOS\\embryoCropUI\\embryoCropUI.exe). Double click to launch (or chose ‘open with’ → terminal) and run the embryoCropUI executable.  
\*\*Note- MacOS distribution is only compatible with Mac OS X 10.11 and higher\*\*.
2. Once the program has started, the following window will appear:

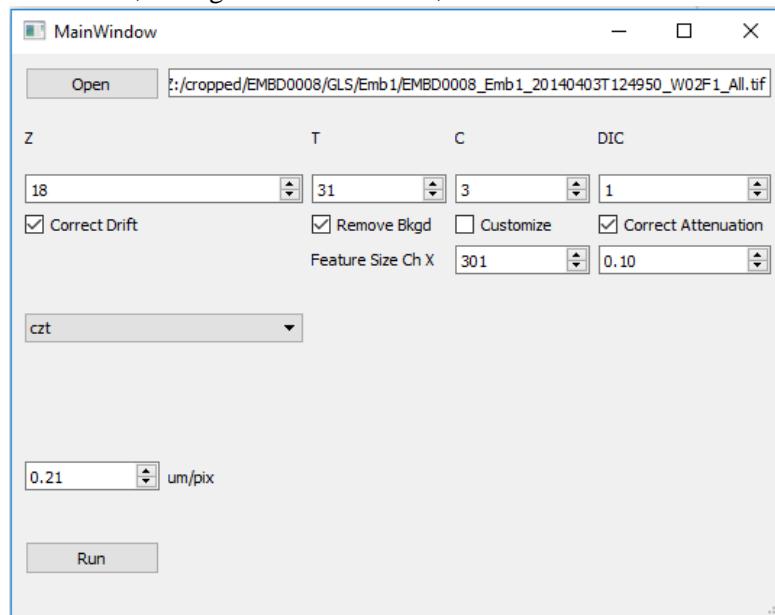


3. Select the “Open” button at the top of the window to load the specific image that you wish to crop. Should you be cropping an image series, with multiple dimensions (i.e. z, time, or channel), simply load the first image in the series within the folder. \*Please make sure only images from one image series are present in this folder, otherwise the image series’ will be loaded in tandem.\*
4. Once you have loaded the desired images, you will need to specify the following information:
  - a. Number of **Z slices (Z)**
  - b. Number of **Time points (T)**
  - c. Number of **Channels (C)**
  - d. The channel that corresponds to **DIC or brightfield** (first=1, second=2, etc)

For example: our imaging protocol was 18 z-steps, imaged for 31 time-points in 3 channels with the DIC channel being the first channel imaged. The window should look like this:

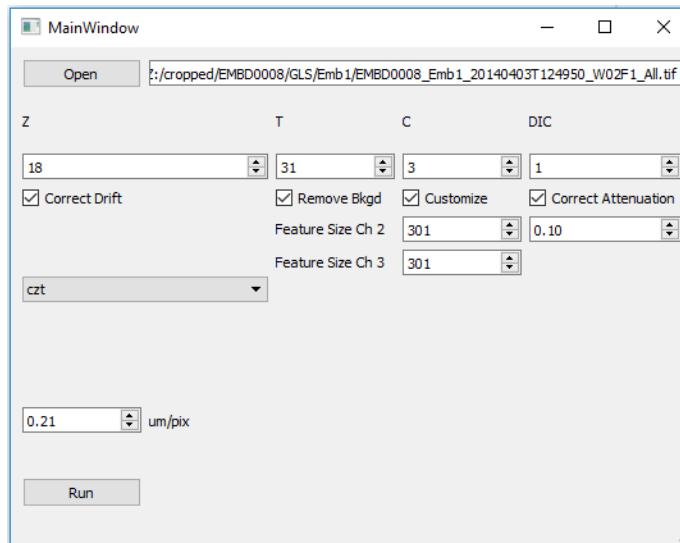


- Now that you have your images loaded and specified the image parameters, you must choose what processing you would like to do alongside the embryo cropping. The program gives you the option to perform **Drift Correction, Background Subtraction, and Attenuation Correction**. Background Subtraction and Attenuation Correction must be done in conjunction with each other. The below image shows an image that will be going through Drift Correction, Background Subtraction, and Attenuation Correction.



- When selecting Background Subtraction and Attenuation Correction, specify parameters for each to guide the processing. For *Background Subtract*, define a feature size (*odd numbered*) that reflects the level of detail you wish to resolve, larger feature size equates to more detail. For *Attenuation Correction*, you need to input a value from 0-1. This value represents the percent of original intensity that remains at the furthest distance through the object being imaged.

- b. As Shown below, you have even greater options to customize Background subtraction. By selecting Customize, you will be able to define a feature size for different channels.



c.

7. Next, specify the **order** in which the images were collected (i.e. **channel-z-time (czt)**, or **z-channel-time (zct)**)  
 8. Specify the **microns per pixel of the camera being used** for the images.  
*a. note that failure to properly define pixel size will result in poor image cropping!!*  
 9. Select **Run** at the bottom left corner and the program  
 When the cropping and processing of your images has completed, the cropped versions will be saved in a new subfolder labeled “crop” in the same folder as the uncropped images.

\*\*Two formats of **test files** are available in the repository- download and unzip. We recommend testing one or both of these to ensure the program is functioning properly on your system:

- i. TESTME2\_BGLI140\_1t\_a1.tif- a compiled multi-tif format
    1. Load file in the ‘open’ field. Set Z=1, T=6, C=3, DIC=1 and use the default settings for all other fields. Click Run. If successful, a message will appear at the bottom of the GUI window that says ‘embryos saved’ and it will generate a folder in the same location as the test file labeled “crop”; this should contain 4 embryos.
  - ii. Test\_field- a folder containing an image series
    1. Load the first image in the test\_field folder into the ‘open’ window. Set Z=18, T=4, C=3, DIC=3, change the pixel size to 0.26um/pix. Click Run. a message will appear at the bottom of the GUI window that says ‘embryos saved’ and it will generate a folder in the same location as the test file labeled “crop”; this should contain 2 embryos.
- \*These files should crop in seconds to minutes, but larger image sequences may take some time. The bottom corner of the GUI window will read-out what the program is doing (“Loading images”, “cropping”, or “embryos saved”). If an error occurs, the message will appear here.

## 2. Running OpenandCombine\_embs (Fiji processing for viewing):

1. This Fiji Script “OpenandCombine\_embs” combines all the images for a specific condition and strain into one easy to view Fiji file. Requires installation of **ImageJ**. Note that this program runs according to our file structure and may need to be modified to work with your

file structure (see our file structure below as a guide). To prepare to run this script, you must know the following information:

- The location where images were stored following cropping.
- The desired location for saving the Fiji files after processing.
- The Target Name for the specific condition you wish to process.
- The Experiment Folder Name.

\*for reference, our file location structure looks like this:

Z:\cropped\Target\Strain\Emb#\Target\_Emb#\_Experiment Folder Name\_W##F#\_T##\_Z##\_C#.tif  
i.e. Z:\cropped\EMBD0002\MS\Emb1\EMBD0002\_Emb1\_20140402T140154\_W06F2\_T01\_Z01\_C1

- Open ImageJ, drag our .ijm Script file to the ImageJ bar. Once the script is open, locate lines 28-31, they are shown below. Within these lines you will fill in the Information you gathered above.

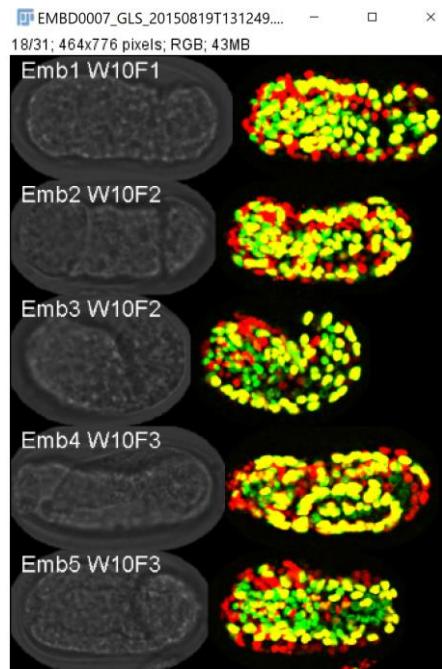
```

27 }
28 RNAL = newArray("EMBD0000/", "EMBD0052/", "EMBD0053/", "EMBD0054/");
29 date = newArray("20140110T153253");
30 folder = "Z:/cropped/";
31 folderOut2 = "Z:/EMBD_fiji_processed/";
32 fSize = 24;
33 setFont( "SansSerif", fSize);
34 for (r=0; r<date.length; r++){
35 for (l=0; l<RNAL.length; l++){
36   RNA = RNAL[l];
37   listStrains = getFileList(folder+RNA);
38   for (j = 0; j < listStrains.length; j++){
39     if (listStrains[j] == "GLS/"){

```

- In Line 30, input the folder location of the images you wish to process in quotations.
  - In Line 31, input the desired save location in quotations.
  - In Line 28, input the Target name for the images you wish to process. Each Target name must be put in the format shown above, with a forward slash at the end and in quotations.
  - In Line 29, input the Image Folder Name in quotations.
- Press Run at the bottom left corner of the script window. The program will then begin processing your images and compiling them into one file. Once the Program has finished the files will be open allowing you to review them. The files can be closed without saving, as the program has already saved them during processing.

Example:



## Github Repository ([https://github.com/renatkh/embryo\\_crop.git](https://github.com/renatkh/embryo_crop.git))

- **embryoCropUI.py**
- **screenCrop.py**
- **README file**

## SETUP

Both programs (**embryoCropUI.py** and **screenCrop.py**) use specific versions of Python and Python modules, thus configuring an appropriate environment is essential for the programs to run. We recommend and provide instructions for installation of **Git**, **Anaconda (includes Python3)**, and **PyCharm** to enable proper environment configuration (detailed instructions below). *\*\*Note- MacOS distribution of embryoCropUI.UI is only compatible with Mac OS X 10.11 and higher\*\**

## DETAILED INSTRUCTIONS

### 1. Configure your environment

#### Clone repository with GIT

1. If you don't already have GIT installed, go to <https://git-scm.com/download/>. You may need to enable security settings to be sure it will download.

2. Check install by going to terminal or command prompt and enter:

> **git --version**

*\*if installed a version will be listed in the terminal*

3. Clone repository:

>**git clone https://github.com/renatkh/embryo\_crop.git**

4. Check in your home directory to ensure that it was properly downloaded.

#### Install Visual Studio (WINDOWS ONLY):

1. Go to [www.visualstudio.microsoft.com/downloads](http://www.visualstudio.microsoft.com/downloads) and **download Visual Studio**. This contains C++ tools, which are required for proper setup of the virtual environment with anaconda .yml files.
2. Select C++ tools
3. Install

#### Setup virtual environment with Anaconda

1. If you don't already have Anaconda, go to [www.anaconda.com/download/](http://www.anaconda.com/download/) and Download **Anaconda3** (python3.7 version), launch anaconda setup and click through default options to install.

2. set environment variables and add conda to the path:

a. Find **conda.exe** location in **Anaconda Prompt**:

WINDOWS: Go to **windows button-> Anaconda3-> Anaconda Prompt**

MacOS: **Anaconda3-> Anaconda Prompt**

At the prompt type in

> **where conda**

```
Anaconda Prompt  
  
(base) C:\Users\rebec>where conda  
C:\Users\rebec\Anaconda3\Library\bin\conda.bat  
C:\Users\rebec\Anaconda3\Scripts\conda.exe
```

Find the location where conda.exe is located (ignore the .bat location) so you can add this location to environmental variables (this can be done within Anaconda Prompt).

In this case it is **C:\Users\rebec\Anaconda3\Scripts**

*But obviously **this will be specific to your system**, so please edit the path appropriately!*

For this example, we need to add both paths:

**C:\Users\rebec\Anaconda3**

**C:\Users\rebec\Anaconda3\Scripts**

b. Add to environment variables. To do this, type

> **SETX PATH "%PATH%;C:\Users\rebec\Anaconda3;C:\Users\rebec\Anaconda3\Scripts"**

```
(base) C:\Users\rebec>SETX PATH "%PATH%;C:\Users\rebec\Anaconda3;C:\Users\rebec\Anaconda3\Scripts"  
  
WARNING: The data being saved is truncated to 1024 characters.  
  
SUCCESS: Specified value was saved.
```

### 3. Close **Anaconda Prompt**

3. Go to **system terminal or command prompt** (*not anaconda prompt*) and check to be sure that the **conda** command works.

> **conda**

*\*this should return information about **conda** functionality. If it does not, you have not successfully added path environmental variables.*

4. Configuring the environment in command line/terminal:

- Navigate to the location where embryo\_crop repository was saved.  
i.e. for example, it is saved here: C:\Users\rebec\embryo\_crop

so at the prompt:

> **cd C:\Users\rebec\embryo\_crop**

- Create new conda environment from .yml file

Once inside the directory, create the environment:

-for Windows: > **conda env create -f environment\_win.yml**

-for MacOS:> **conda env create -f environment\_mac.yml**

This step will take a few minutes to solve the environment.....

```
C:\Users\rebec\embryo_crop>conda env create -f environment_win.yml
Solving environment: done
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
Collecting pyqt5==5.11.3 (from -r C:\Users\rebec\embryo_crop\condaenv.lmfjd3fz.requirements.txt)
  Using cached https://files.pythonhosted.org/packages/a7/2d/d2c989006c86ae98ed230c28c3e0dd7fa037...
Collecting pyqt5-sip==4.19.13 (from -r C:\Users\rebec\embryo_crop\condaenv.lmfjd3fz.requirements.txt)
  Using cached https://files.pythonhosted.org/packages/46/86/0e35563d0c67c3f6b50e344624b87bfa7e72...
Collecting tifffile==0.15.1 (from -r C:\Users\rebec\embryo_crop\condaenv.lmfjd3fz.requirements.txt)
  Using cached https://files.pythonhosted.org/packages/1f/a1/4055cd679081cb4c1e40aa7648adb12574bf...
Requirement already satisfied: numpy>=1.8.2 in c:\users\rebec\anaconda3\envs\embryocrop\lib\site-
Building wheels for collected packages: tifffile
  Running setup.py bdist_wheel for tifffile ... done
  Stored in directory: C:\Users\rebec\AppData\Local\pip\Cache\wheels\22\5b\8e\9bc85b5dfc1cc91b84...
Successfully built tifffile
mkl-random 1.0.1 requires cython, which is not installed.
Installing collected packages: pyqt5-sip, pyqt5, tifffile
  The scripts pylupdate5.exe, pyrc5.exe and pyuic5.exe are installed in 'C:\Users\rebec\Anaconda...
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-war...
Successfully installed pyqt5-5.11.3 pyqt5-sip-4.19.13 tifffile-0.15.1
#
# To activate this environment, use:
# > activate embryocrop
#
# To deactivate an active environment, use:
# > deactivate
#
# * for power-users using bash, you must source
#
```

5. When finished, you can continue in command line to run embryoCropUI (below) or switch to an IDE to run screenCrop or embryoCropUI.

a. *To continue in command line (for embryoCropUI):*

Activate the environment according to the instructions listed in the terminal.

For Windows:

**> activate embryocrop**

For MacOS:

**> source activate embryocrop**

Once activated, you can run python programs by calling the program in command line

**>python embryoCropUI.py**

\*this will launch the GUI window- please follow the instructions for the GUI use (in the readme file).

When finished:

Close GUI

**>deactivate**

## Configure Environment in IDE (Setup in PyCharm)

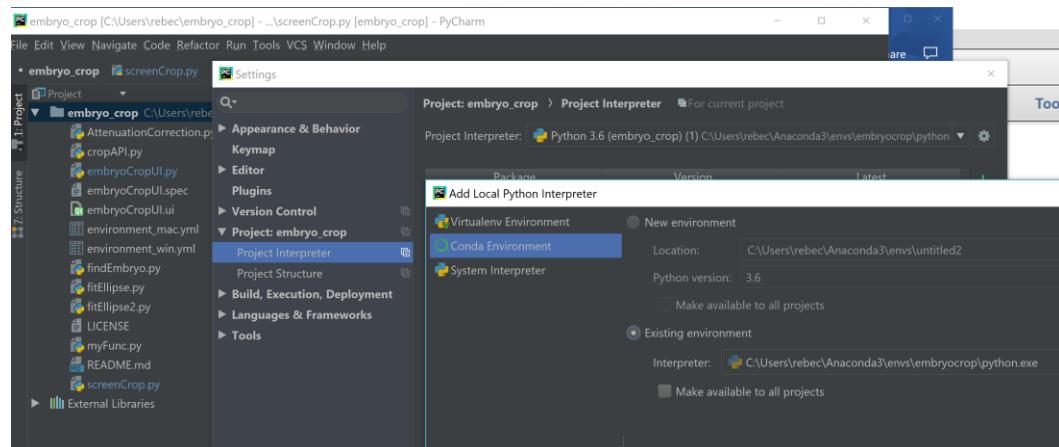
To access the code directly, which is necessary for **screenCrop.py** functionality, open the environment in your favorite **Integrated Development Environment (IDE)**. There are many IDEs that are commonly used for coding in Python. An overview of features and limitations for popular options can be found here: <https://stackoverflow.com/questions/81584/what-ide-to-use-for-python>.

We include instructions for installing and running with **PyCharm**, however if you already have a different IDE that is compatible with Python (i.e. Eclipse) it should work fine. Note that Jupyter notebooks currently DOES NOT support GUIs, so our embryoCropUI will not run properly in this environment.

- a. To get **PyCharm**, go to: <https://www.jetbrains.com/pycharm/download/#section=windows>. Select your operating system and click the black download button under “Community”.
  - i. For Windows- run the PyCharm-\*.exe file and follow the installation wizard.
  - ii. For MaOSc, open the PyCharm-\*.dmg package and drag PyCharm to the Application folder.
- b. When you start PyCharm for the first time, you will be prompted with **a complete installation** dialog box. Click **Do not import settings**.
- c. Select your preferred user interface theme [Default or Darcula (black background)]
- d. You will be prompted to install any additional plugins (not necessary)
- e. Now you should be set up and greeted with a “welcome screen” where you can create, open, or check out a project.

### In PyCharm:

1. **File > Open > embryo\_crop**
2. Configure environment. Go to **file > settings > project:embryo\_crop > project interpreter > add local** (select conda) > **existing environment >**  
select the newly generated conda env from within the Anaconda3 envs folder:  
**...Anaconda3\envs\embryocrop\python.exe**



3. From here, you should be able to run programs using the PyCharm ‘run’ button (program instructions below). If this doesn’t properly structure the environment, it may crash. If this happens, you can access the *terminal window within PyCharm* and activate the environment this way:
  - a. For Windows:  
> **activate embryoCrop**
  - b. For Mac OS:  
> **source activate embryoCrop**
4. Once the environment is activated, you can run the program via the *terminal within PyCharm*:

```
> python screenCrop.py
or > python embryoCropUI.py
```

\*\* note that the screenCrop program will need to be modified to work with your file structure!!See instructions below.

```
C:\Users\rebec\embryo_crop>activate embryocrop

(embryocrop) C:\Users\rebec\embryo_crop>python screenCrop.py
STARTED!!!
screenCrop.py:36: DeprecationWarning: 'U' mode is deprecated
    csvFile = csv.reader(open(fileName, 'rU'), delimiter=',') # universal
```

4: Run 6: TODO 9: Version Control Python Console Terminal

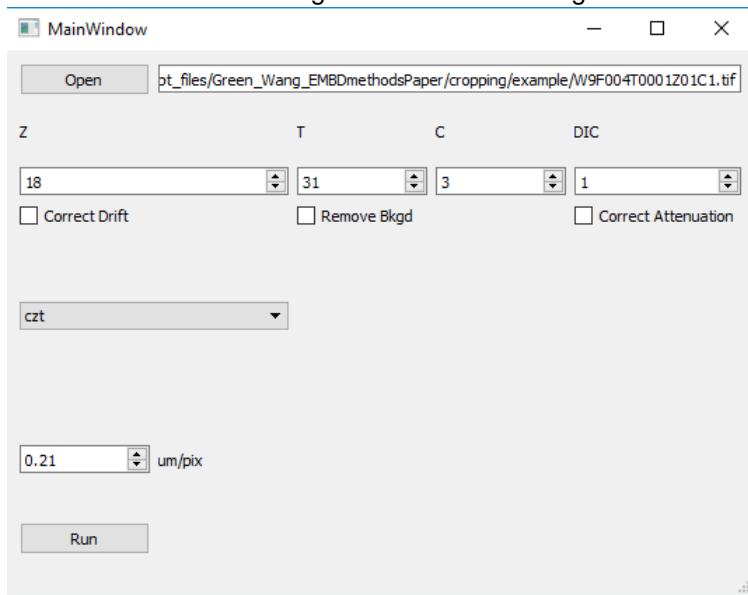
When finished:

>**deactivate**

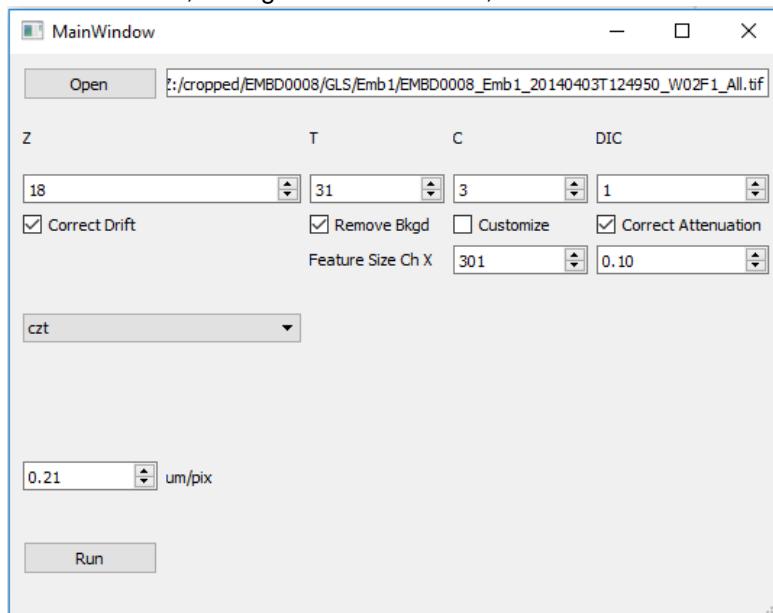
## 2. Running embryoCropUI.py GUI with Python

1. In PyCharm (or your preferred IDE) double click on the **embryoCrop** folder and locate the file that says **embryoCropUI.py** (DO NOT open embryoCropUI.ui). The code will appear in the workspace.
2. If this is the only file open, go to the top right-hand corner and **click the green triangle** to start the run. If multiple files are open, right click and select '**Run embryoCropUI**' to ensure the proper program is run. Alternatively, activate the environment and run from PyCharm terminal, as outlined above.
3. Once the program has started, the following window will appear:
4. Select the “**Open**” button at the top of the window to load the specific image that you wish to crop. Should you be cropping an image series, with multiple dimensions (i.e. z, time, or channel), simply load the first image in the series within the folder. \*Please make sure only images from one image series are present in this folder, otherwise the image series’ will be loaded in tandem.\*
5. Once you have loaded the desired images, you will need to specify the following information:
  - e. Number of **Z slices (Z)**
  - f. Number of **Time points (T)**
  - g. Number of **Channels (C)**
  - h. The channel that corresponds to **DIC or brightfield** (first=1, second=2, etc)

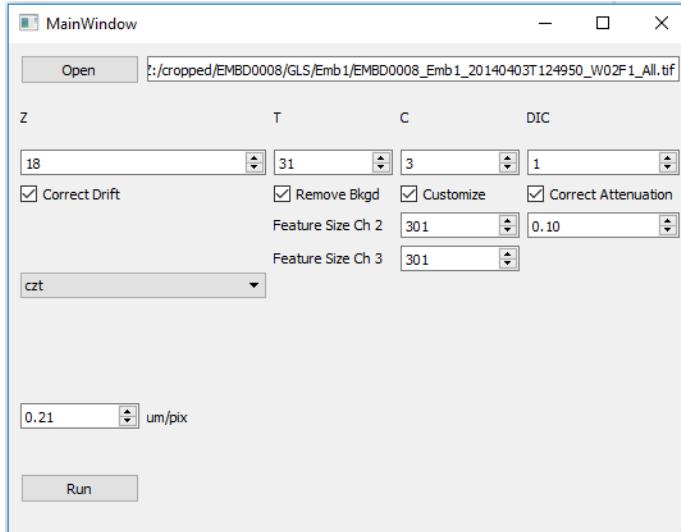
For example: our imaging protocol was 18 z-steps, imaged for 31 time-points in 3 channels with the DIC channel being the first channel imaged. The window should look like this:



6. Now that you have your images loaded and specified the image parameters, you must choose what processing you would like to do alongside the embryo cropping. The program gives you the option to perform **Drift Correction, Background Subtraction, and Attenuation Correction**. Background Subtraction and Attenuation Correction must be done in conjunction with each other. The below image shows an image that will be going through Drift Correction, Background Subtraction, and Attenuation Correction.



- d. When selecting Background Subtraction and Attenuation Correction, specify parameters for each to guide the processing. For *Background Subtract*, define a feature size (*odd numbered*) that reflects the level of detail you wish to resolve, larger feature size equates to more detail. For *Attenuation Correction*, you need to input a value from 0-1. This value represents the percent of original intensity that remains at the furthest distance through the object being imaged.
- e. As shown below, you have even greater options to customize Background subtraction. By selecting *Customize*, you will be able to define a feature size for different channels.



f.

7. Next, specify the **order** in which the images were collected (i.e. **channel-z-time (czt)**, or **z-channel-time (zct)**)

8. Specify the **microns per pixel of the camera being used** for the images.

*a. failure to properly define pixel size will result in poor image cropping!!*

10. Select **Run** at the bottom left corner and the program

When the cropping and processing of your images has completed, the cropped versions will be saved in a new subfolder labeled “crop” in the same folder as the uncropped images.

\*\*We make available two formats of test files for testing embryoCropUI.py. This folder is too large for Github requirements and thus is stored on the Zenodo repository (<https://zenodo.org/record/1475442#.W9jvApNKiUI>). Download and unzip. We recommend testing one or both to ensure the program is functioning properly on your system:

iii. TESTME2\_BGLI140\_1t\_a1.tif- a compiled multi-tif format

1. Load file in the ‘open’ field. Set Z=1, T=6, C=3, DIC=1 and use the default settings for all other fields. Click Run. If successful, a message will appear at the bottom of the GUI window that says ‘embryos saved’ and it will generate a folder in the same location as the test file labeled “crop”; this should contain 4 embryos.

iv. Test\_field- a folder containing an short image series

1. Load the first image in the test\_field folder into the ‘open’ window. Set Z=18, T=4, C=3, DIC=3, change the pixel size to 0.26um/pix. Click Run. a message will appear at the bottom of the GUI window that says ‘embryos saved’ and it will generate a folder in the same location as the test file labeled “crop”; this should contain 2 embryos.

\*These files should crop in seconds to minutes, but larger image sequences may take some time. The bottom corner of the GUI window will read-out what the program is doing (“Loading images”, “cropping”, or “embryos saved”). If an error occurs, the message will appear here.

### 3.Running screenCrop.py

This software allows you to batch crop many files at once, but it is less user friendly and has not been optimized across platforms. *It was designed to function with data output from CV1000 imaging systems*; if you have a CV1000 and follow the instructions below, it should work seamlessly. If you have another system, modifications to the code will likely be required and someone with Python experience will be

needed. In the event that it is needed, we outline key elements of the code and our file structure to guide such efforts. Successful bulk cropping requires:

1. A **reference .csv** file that contains essential image information, which is called by our Python software
2. **Properly named files**

#### CSV:

The CSV file contains information that will be called on during processing or used to generate the cropped file path. Below is an example .csv file that is compatible with our programs. Formatting your .csv file in the same way will ensure your data will go through our programs with minimal issues.

Experiment	Experiment Folder Name	Post-scan folder name	Well designation	Target	strain	Plate Coordinate
EMBD_112013	20131120T160955	20131121T104921	Well001	EMBD0000	GL	C3
	20131120T160955	20131121T104921	Well005	EMBD0000	GLS	D3
	20131120T160955	20131121T104921	Well009	EMBD0000	M	E3
	20131120T160955	20131121T104921	Well013	EMBD0000	MS	F3
	20131120T160955	20131121T104921	Well002	EMBD0002	GL	C4
	20131120T160955	20131121T104921	Well006	EMBD0002	GLS	D4
	20131120T160955	20131121T104921	Well010	EMBD0002	M	E4
	20131120T160955	20131121T104921	Well014	EMBD0002	MS	F4
	20131120T160955	20131121T104921	Well003	EMBD0003	GL	C5
	20131120T160955	20131121T104921	Well007	EMBD0003	GLS	D5
	20131120T160955	20131121T104921	Well011	EMBD0003	M	E5
	20131120T160955	20131121T104921	Well015	EMBD0003	MS	F5
	20131120T160955	20131121T104921	Well004	EMBD0015	GL	C6
	20131120T160955	20131121T104921	Well008	EMBD0015	GLS	D6
	20131120T160955	20131121T104921	Well012	EMBD0015	M	E6
	20131120T160955	20131121T104921	Well016	EMBD0015	MS	F6

Rundown of .CSV file contents:

- Experiment: arbitrary name given to each experiment (not important for software functionality, but this column needs to be maintained)
- Experiment Folder Name: name of folder in where specific experiments images are stored. We prefer Date/Time file name, though any name will suffice (do not include spaces or disallowed characters, as the contents of this cell are added to the file path).
- Post-Scan Folder Name: name of post- scan (10x data) folder. Not important for software functionality, but this column needs to be maintained. You can populate this with 'empty'.
- Well Designation: Well numbers as determined by CV1000 software adhering to the Well### regime.
- Target: Experimental conditions (e.g. RNAi condition), we use a blinded, unique identifier system (EMBD####) for our experimental conditions, though this is not necessary. Output files will be saved according to this name, so do not include spaces or disallowed characters.
- Strain: Specific strain used in experiment. Scaling and background subtraction is applied differently depending on the strain used.
- Plate Coordinate: Coordinates from 384 well plate (for reference, not used by program).

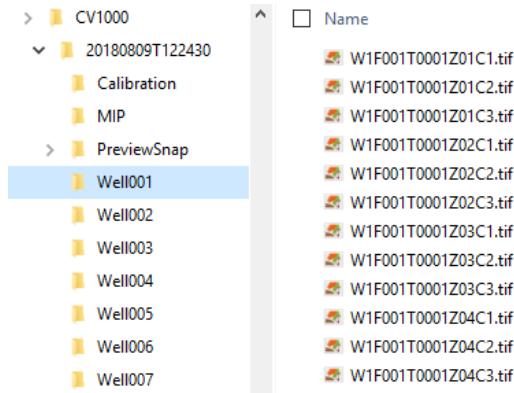
#### Properly naming files:

##### *Input path:*

The path to access the raw data files is referred to in **screenCrop.py**, based on the information in the CSV file (highlighted in red below). Our data is structured such that multiple point visits are contained within each well folder and all of the image files are listed within that well (not in separate subfolders per point visit). Image files have been automatically named according to CV1000 software image naming conventions, as follows:

Z:\CV1000\Experiment Folder Name\Well designation\W##F##T###Z##C#.tif

i.e. Z:\CV1000\20180809T122430\Well001\W1F001T0001Z01C1.tif.

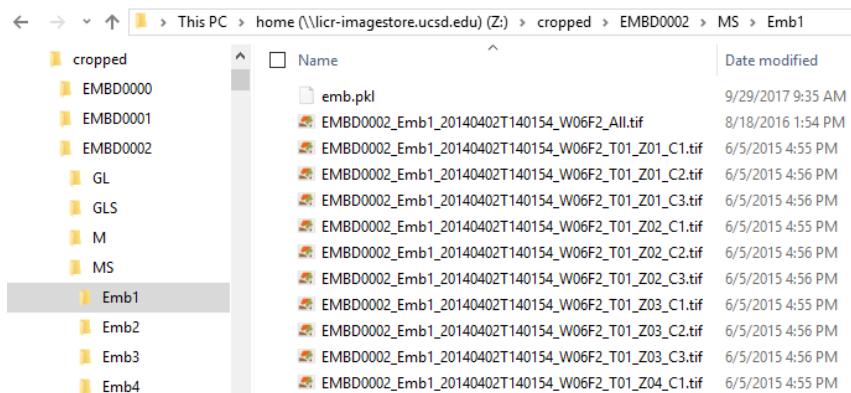


*Output path:*

The output path specified in **screenCrop.py** is also generated based on information in the CSV file (highlighted in red below). Our saving regime uses our naming scheme (Emb#) for outer folders representing data for each embryo and this contains individual tifs that are named as follows:

Z:\cropped\Target\Strain\Emb#\Target\_Emb#\_Experiment Folder Name\_W##F#\_T##\_Z##\_C#.tif

i.e. Z:\cropped\EMBD0002\MS\Emb1\EMBD0002\_Emb1\_20140402T140154\_W06F2\_T01\_Z01\_C1



## Cropping Your Raw Images

Using **ScreenCrop.py**, you will be able to crop all your images from a folder. The program crops each image by fitting an ellipse to each embryo at the fourth time point.

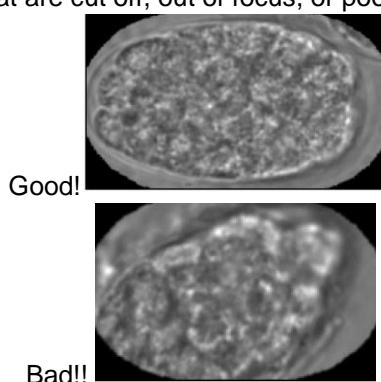
1. Open PyCharm, or other IDE, load the embryo\_crop repository and locate the program screenCrop.py. It is important to know the following Information and fill it in at the specified line:
  - a. loadFolder (line 9): The drive on which the files are stored (e.g. Z:/ , D:// etc.)
  - b. date (line 7): this is the file referred to as Experiment Folder Name in the CSV
  - c. trackingFile (line 11): the path to the CSV file in which experiment information is stored
  - d. z (line 13): The number of z planes
  - e. nT (line 14): Number of timepoints
  - f. nWells (line 19): the number of wells used
  - g. pointVisits (line 20): the maximum number of point visits (per well)

```

4 Loads embryo images generated by CV1000 and crops/orients and saves separate embryos based on DIC (C3) central image
5 ''
6
7 date = '20170118T125754'
8
9 loadFolder = 'Z:/'
10 folderIn = loadFolder + 'CV1000/' + date #Input folder
11 trackingFile = 'Z:/Experiment_tracking_sheets/EMBD_fileNames_Tracking_Sheet.csv'
12 aspectRatioFile = 'Z:/cropped/aspects.csv'
13 z = 18 #number of z planes
14 nT = 31 #number of time points
15 corrDrift = True
16 removeBG = True
17 attCorrect = True
18 apRotate = True
19 nWells = 14#number of wells (14)
20 pointVisits = 4# number of point visits (4)
21
22 import glob, csv, cv2, os, shutil
23 import numpy as np
24 from findEmbryo import showIm
25 from myFunc import clearFolder
26 from cropAPI import cropEmbs
27 import tkMessageBox
28
29 debug = False # use to debug the program
30
31 def getConditions(date, fileName):
32     ''' loads RNAi strains for a specified date from a csv file '''
33     global RNAi, strains
34     csvFile = csv.reader(open(fileName, 'rb'), delimiter=',')
35     csvFile = csv.reader(open(fileName, 'rU'), delimiter=',') #universal
36     fileData=[]
37     for row in csvFile:
38         fileData.append(row[1:-1])
39     myDate = [s for s in fileData if s[0]==date]
40     myDate = sorted(myDate, key=lambda well: well[2])
41     RNAi = [s[3] for s in myDate]
42     strains = [s[4] for s in myDate]
43     return

```

- h. In line 10 find the location currently occupied by 'CV1000/' and input the outer folder used in your file path. To avoid issues, use the following convention: 'XXXXXXX/'.
  - i. In Line 12, input a valid file path for storing aspect ratio data for the cropped.
  - j. In Lines 15, 16, 17, and 18 input True/False for whether you would like your images to go through the following processing:
    - i. Drift Correction (Line 15)
    - ii. Background Subtract (Line 16), feature size should be defined as 41 for GLS strain and 201 for MS strain. Background Subtract must be done in conjunction with Attenuation Correction.
    - iii. Attenuation Correction (Line 17)
    - iv. AP Rotate (Line 18)
2. Once all the changes have been made to tailor the program for your data, you may begin cropping. This is done by selecting the green play icon in the toolbar, this will have a drop down menu where you select "Run As" and then "Python Run". Alternatively, activate the environment and run from PyCharm terminal (described above at the end of the PyCharm section).
3. The Program will then begin cropping your images, this may take a few hours depending on the number of images that need to be processed. Once completed, a series of small windows containing embryo images will open; this will allow you to curate the cropped data before saving (i.e. delete embryos that are cut off, out of focus, or poorly cropped can be deleted).



4. For each image you have three options:
  - a. **Save:** If the image appears to be cropped properly with no areas of interest being cut off, press the space bar to save the image.
  - b. **X:** If the image appears to have areas of interest cut off and you still wish to save the image, press X and the image will be saved with an X in front of the name to separate it from the others.
  - c. **Delete:** If the image is not cropped properly or the embryo is not to your liking, press D to delete the cropped image.
5. Once you have gone through all your images and determined whether you wish to **save, x, or delete** them, the program will then begin to save your images. ***The images will be saved to a subfolder named “Cropped” in the Load Folder that was defined in Line 9 of the program.***