Science Advances

AAAS

advances.sciencemag.org/cgi/content/full/2/8/e1600060/DC1

Supplementary Materials for

Analysis of neural crest–derived clones reveals novel aspects of facial development

Marketa Kaucka, Evgeny Ivashkin, Daniel Gyllborg, Tomas Zikmund, Marketa Tesarova, Jozef Kaiser, Meng Xie, Julian Petersen, Vassilis Pachnis, Silvia K. Nicolis, Tian Yu, Paul Sharpe, Ernest Arenas, Hjalmar Brismar, Hans Blom, Hans Clevers, Ueli Suter, Andrei S. Chagin, Kaj Fried, Andreas Hellander, Igor Adameyko

Published 3 August 2016, *Sci. Adv.* **2**, e1600060 (2016)
DOI: 10.1126/sciadv.1600060

The PDF file includes:

Other Supplementary Material for this manuscript includes the following:
(available at advances.sciencemag.org/cgi/content/full/2/8/e1600060/DC1)

- movie S1 (.avi format). Live imaging of genetically traced neural crest–derived progenies in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 30 and 56 hpf, ventral view.
- movie S2 (.avi format). Live imaging of translocating ectomesenchymal clusters in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 39 and 52 hpf, ventral view.

- movie S3 (.avi format). Live imaging of genetically traced neural crest–derived progenies in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 30 and 88hpf, ventral view.
- movie S4 (.avi format). Live imaging of translocating ectomesenchymal clusters in Sox10-*CreERT2/Ubi:zebrabow-S* zebrafish larvae between 39 and 52 hpf, ventral view.
- movie S5 (.avi format). 3D EdU analysis of *Col2a1aBAC:mCherry* zebrafish larva's entire head at 4 dpf corresponding to Fig. 5 (O to Q).

Supplementary Figures and Legends

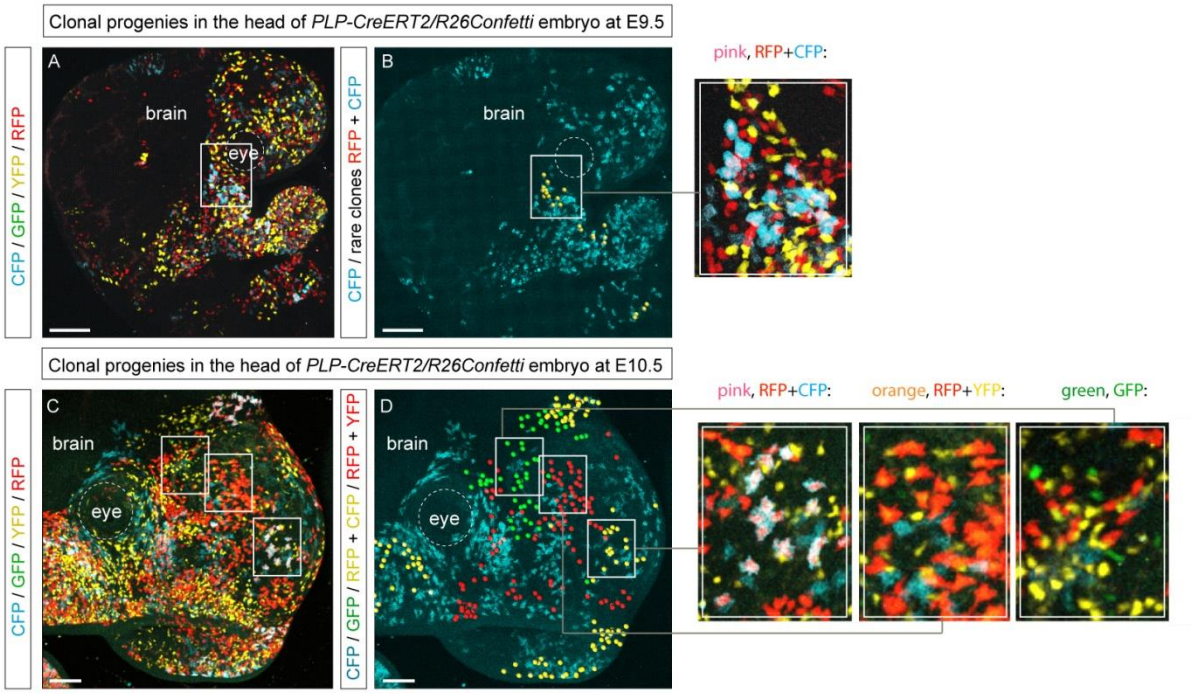


fig. S1. Identification of rare double-color and GFP$^+$ clones in neural crest ectomesenchyme in E9.5 to E10.5 embryonic faces. (**A-B**) E9.5 mouse embryo head of *PLP-CreERT2/R26Confetti* embryo with rare CFP$^+$/RFP$^+$ clone. (**C-D**) E10.5 mouse embryo head of *PLP-CreERT2/R26Confetti* embryo with rare CFP$^+$/RFP$^+$, RFP$^+$/YFP$^+$ and GFP$^+$ clones. Scale bars are 200 µm.

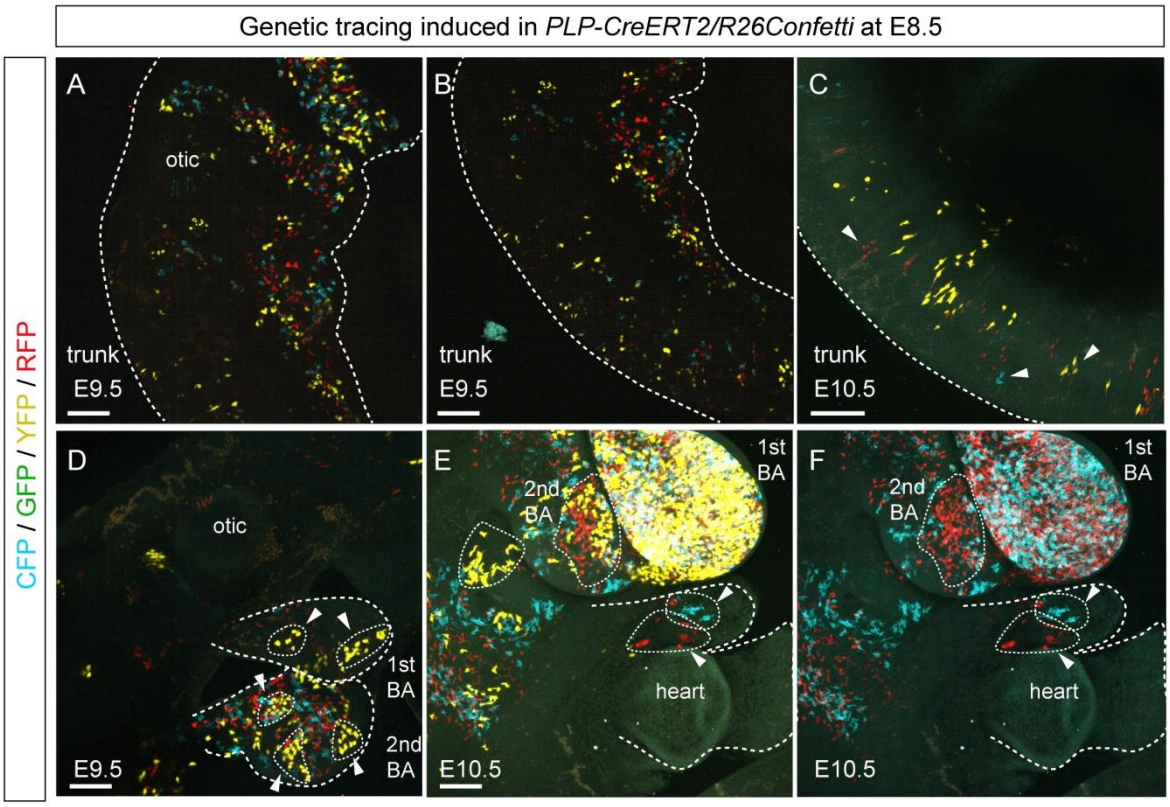


fig. S2. Clonal mixing and distribution of NCC-derived clones in the embryonic trunk and head through the development. (**A-C**) Examples of tracing migratory neural crest in the trunks of *PLP-CreERT2/R26Confetti* embryos injected with tamoxifen at E8.5 and analyzed at E9.5 (A-B) and E10.5 (C). Note the absence of large clonal clusters among migrating neural crest. (**D-F**) Same type of genetic tracing showing branchial arches and cardiac neural crest analysed at E9.5 and E10.5. Note efficient clonal mixing and formation of clonal clusters already at E9.5 (D). Note the presence of two cardiac neural crest clones (red and blue, pointed out by arrowheads) in the area of developing heart. (**G**) Individual clones occupy defined areas in E16.5 genetically traced mandible (BA1). BA – branchial arch. Scale bars are 200 µm.

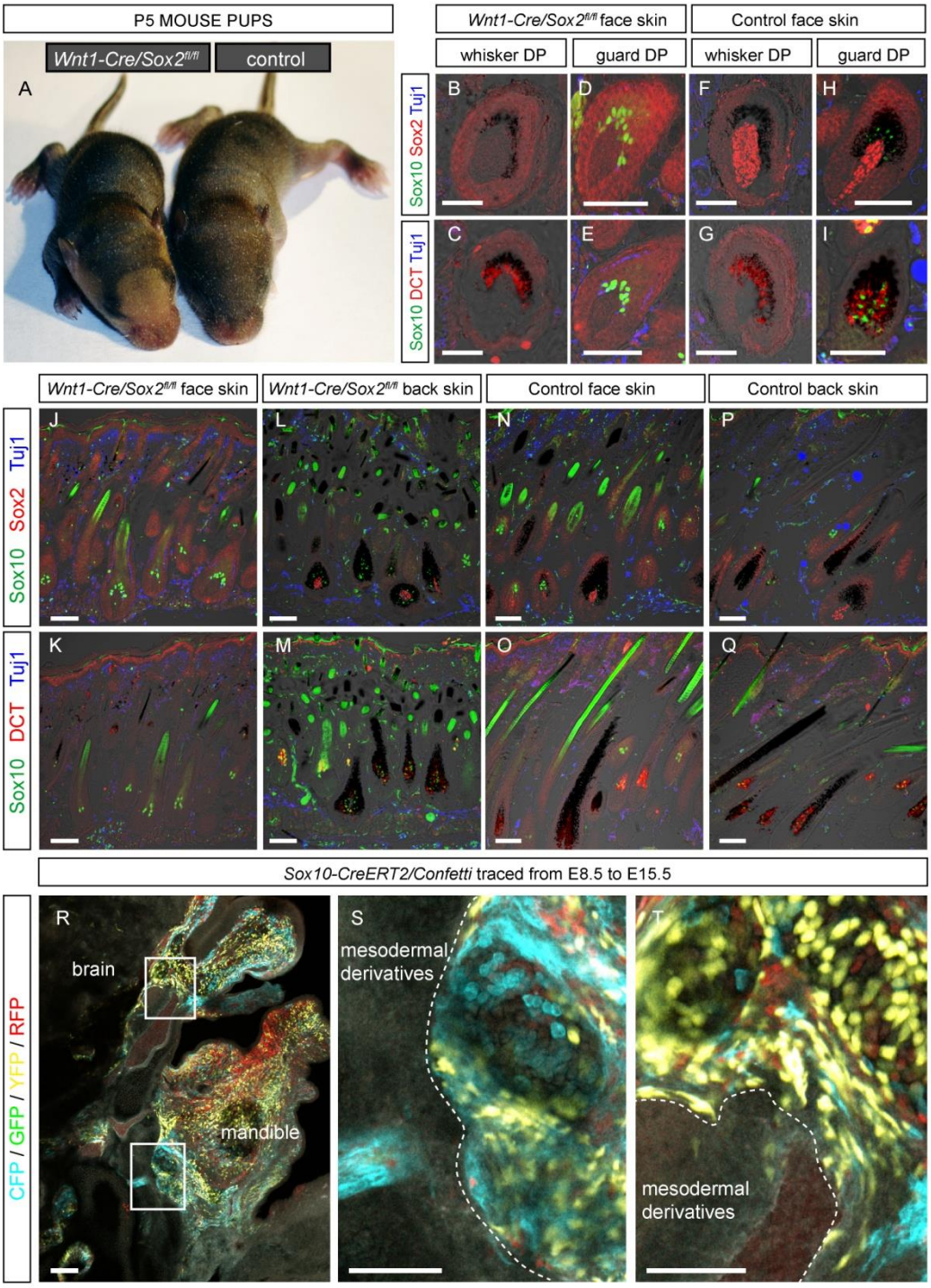
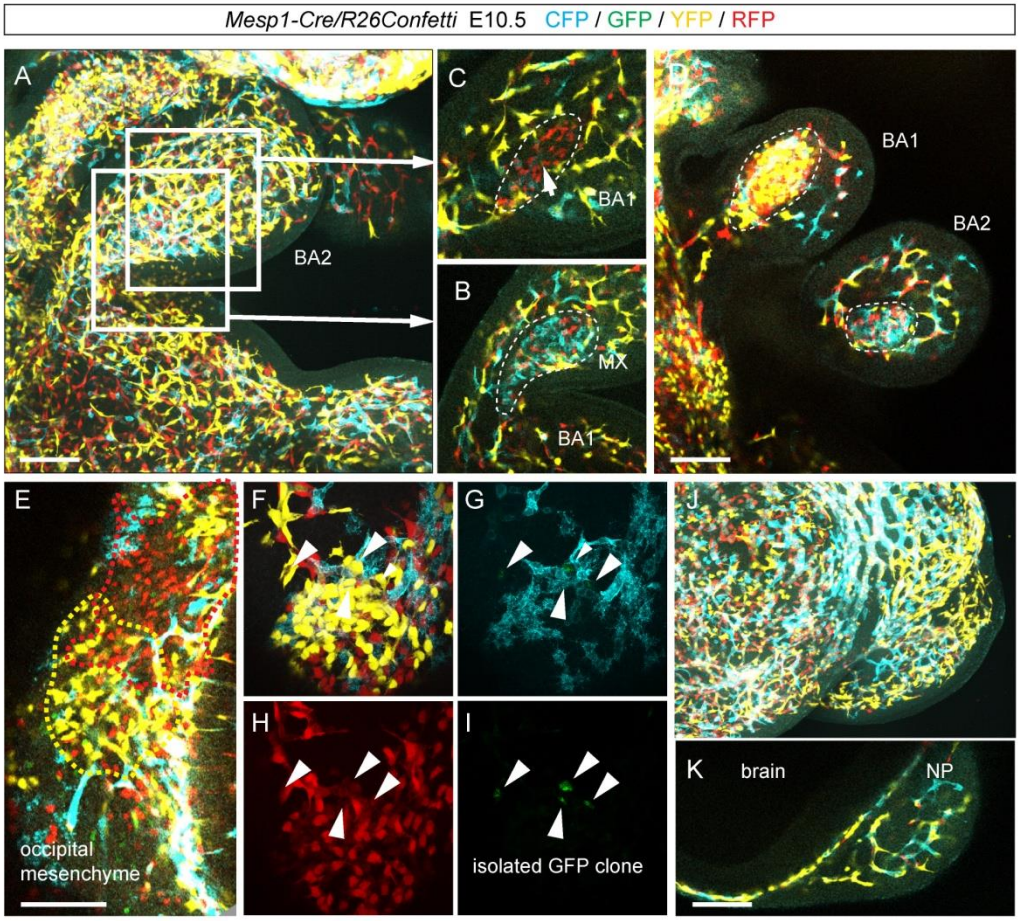


fig. S3. Defined borders between mesoderm- and neural crest–derived progenies at postnatal and embryonic stages. (**A-Q**) Genetic deletion of Sox2 from the neural crest cells using Wnt1-Cre mouse line affects the pigmentation of the skin with neural crest-derived dermis and depicts the sharpness of the borders that are, due to the stable tissue dynamics,

maintained till P5 of postnatal life. (A) Sox2 conditional KO mutant P5 pup with defects in pigmentation on the left next to the littermate control on the right. (B-I) Difference in pigmentation of whisker follicle and guard hair follicle between Sox2 conditional KO mutant and control conditions. (J-M) Overview of the facial skin from Sox2 conditional KO mutant and control pups. Note almost complete lack of pigmentation in Sox2 KO condition. (N-Q) Overview of the back skin from Sox2 conditional KO mutant and control pups. Note the absence of difference in pigmentation of back skin from mutant and control embryos since the dermis of the back skin is not neural-crest derived. (**R-T**) Neural crest genetic tracing induced in E8.5 Sox10-CreERT2/R26confetti embryo analyzed at E15.5 shows clear borders between neural crest-derived areas and those of mesodermal origin (non-traced). Scale bars are 100 µm.


Mesp1-Cre/R26Confetti E10.5 CFP / GFP / YFP / RFP
A
BA2
C
BA1
B
MX
BA1
D
BA1
BA2
E
occipital mesenchyme
F
G
H
I
isolated GFP clone
J
K
brain
NP

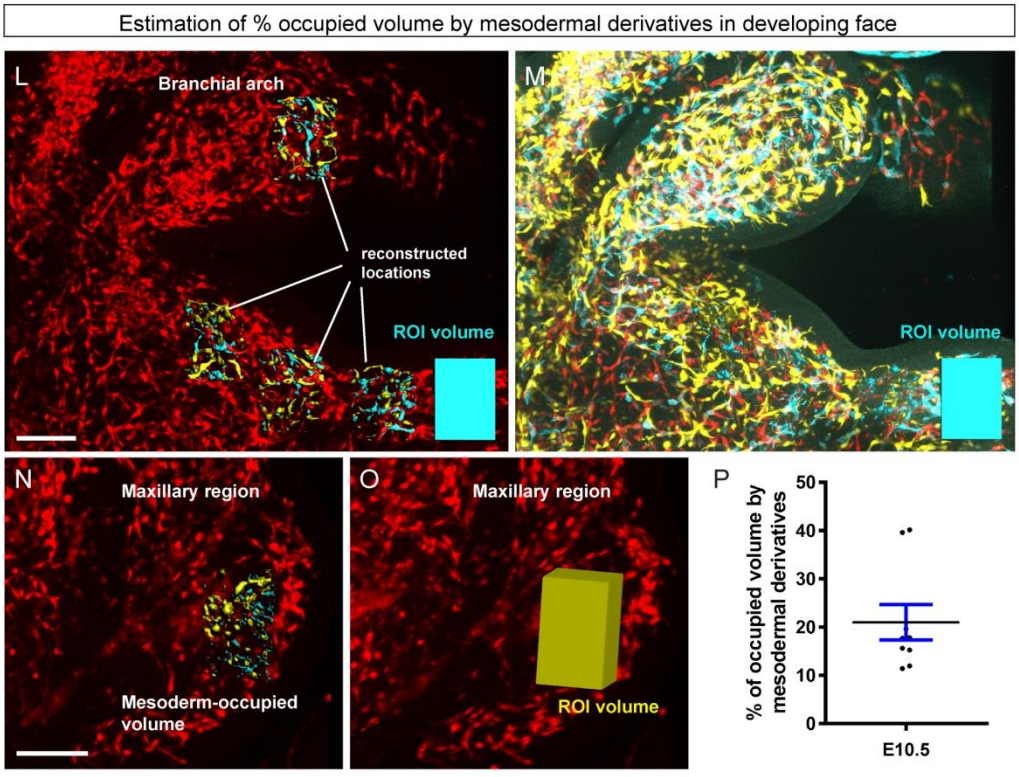

Estimation of % occupied volume by mesodermal derivatives in developing face
L
Branchial arch
reconstructed locations
ROI volume
M
ROI volume
N
Maxillary region
Mesoderm-occupied volume
O
Maxillary region
ROI volume
P
% of occupied volume by mesodermal derivatives
50
40
30
20
10
0
E10.5

fig. S4. Genetic tracing of mesoderm-derived mesenchymal progenies reveals similarities with the neural crest–derived ectomesenchyme. (**A-K**) Genetic tracing in MesP1-Cre/R26Confetti embryos analyzed at E10.5. MIP images of confocal stacks. (A-D) Branchial arches with shown angiogenic and non-angiogenic mesodermal derivatives. Non-angiogenic derivatives of paraxial mesoderm are outlined with a dashed line in (B, D). Note that non-angiogenic derivatives localizing in the center of the arch may contain only one clonal color, for example, RFP (pointed out by arrow in B). BA1/2 – branchial arches 1 and 2. (E-I) Multiple spatially overlapping clonal envelopes of mixing occipital mesoderm-derived mesenchymal clones. Arrowheads in (F-I) point out at the cells of the rare GFP^+ mesenchymal clone. (J-K) In the anterior face mesodermal derivatives are represented mostly by developing vessels formed via fusion of clonally heterogeneous angiogenic progenitors. (**L-P**) 3D analysis of mesodermal derivatives in the head represented by angiogenic and muscle progenitors. Note the examples of locations where fluorescent protein-labelled mesodermal cells were reconstructed as isosurfaces within standard region of interest (ROI) volume. Panel (P) shows the graph describing % volume of mesodermal contribution inside ROI standard volume. NP - nasal prominence. Scale bars are 150 μm.

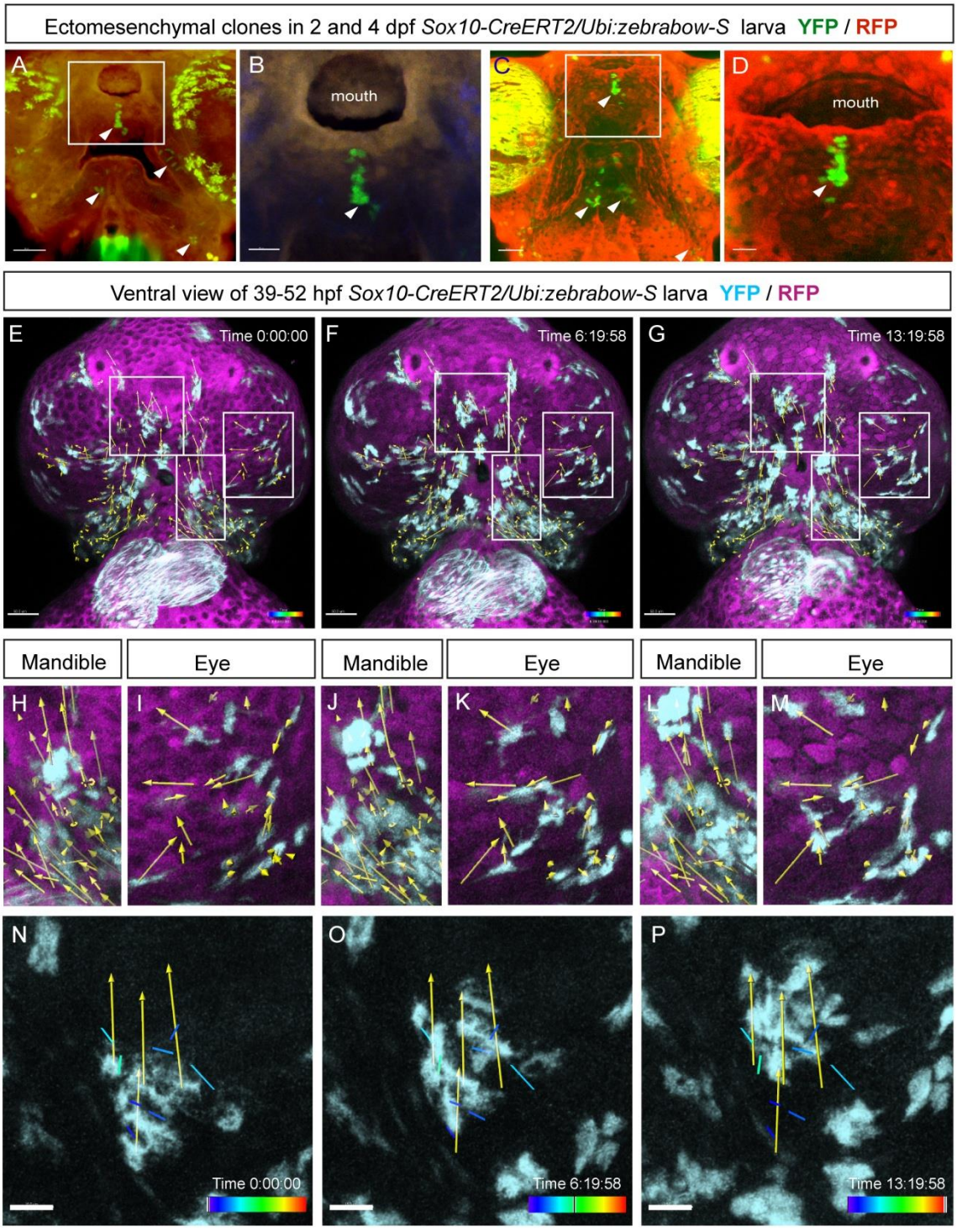


fig. S5. Live imaging of ectomesenchymal clones and progenitors in the eye shows difference between organized crowd movements and individual migrations. (**A-D**) Clonal density tracing of neural crest progeny (4-hydroxytamoxifen administered at 16 hpf) in zebrafish ectomesenchyme of *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larva. Arrows point

out at separate compact groups of YFP^+ cells in the developing head. Note that during two days of development the groups still retain compact structure and defined borders. (**E-P**) Ventral view at YFP^+ectomesenchymal clones and progenitors in the eye moving and proliferating at three different time-points. YFP-expressing recombined cells are shown as cyan for better visual compatibility with displacement arrows. (E-G) Note coordinated displacement/crowd movement of numerous cells shown by yellow displacement arrows (also see also Supplemental Video Figures). (H-M) Magnified regions from panels (E-G) showing oriented crowd movements/translocations in developing mandible/branchial arch region (H,J,L) as compared to individual cell migrations in the developing eye (I,K,M). The yellow arrows represent displacement lengths from the start to the end point without showing the exact cell track. (N-P) Magnified group of cells in front of stomodeum from outlined area in (E-G) with displacement arrows at three different timepoints. Orientations of cell divisions are shown with bars that are color-coded according to the developmental timing. Note that these ectomesenchymal cells that are relocated as a perfect group without significantly changing the distances between individual cells. Scale bars are 50 μm in A,C,E,G and 50 μm in B,D,N,P.

movie S1. Live imaging of genetically traced neural crest–derived progenies in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 30 and 56 hpf, ventral view. Note the differences in migratory behavior of labelled cell in the eyes and in ectomesenchyme.

movie S2. Live imaging of translocatingectomesenchymal clusters in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 39 and 52 hpf, ventral view.

movie S3. Live imaging of genetically traced neural crest–derived progenies in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 30 and 88 hpf, ventral view. Note the translocation of labelled ectomesenchymal groups together with stomodeum migrating forward and formation of cartilaginous jaw elements seen as cell stacking.

movie S4. Live imaging of translocating ectomesenchymal clusters in *Sox10-CreERT2/Ubi:zebrabow-S* zebrafish larvae between 39 and 52 hpf, ventral view.

movie S5. 3D EdU analysis of *Col2a1aBAC:mCherry* zebrafish larva's entire head at 4 dpf corresponding to Fig. 5 (O to Q). Label-retaining analysis is done at 4 dpf with EdU administered at 48 hpf during 5 min.

Supplementary Materials and Methods

Computational Model

In order to model the process of clonal mixing, we developed a lattice-based stochastic model that accounts for cell division and cell migration. The computational model is an individual based model (IBM), drawing on ideas from probabilistic cellular automata. An overview of IBM modeling for tissue patterning with a review of IBMs as well as the biological conclusions they have led to in areas such as immunology, development, and cancer, can be found in (Thorne et al., 2007). Our model is stochastic because we are interested in modeling the intrinsic, naturally occurring variability that can be expected in the multicellular system arising from for example variation in cell division times, or stochasticity in direction of allocation of daughter cells after cell division. This document describes the details of the model and its implementation. The simulation code is written in Python, relies on the PyURDME package for spatial stochastic simulations (www.pyurdme.org) and it is freely available for downloads from www.github.com/ahellander/multicell under the GPLv3 license.

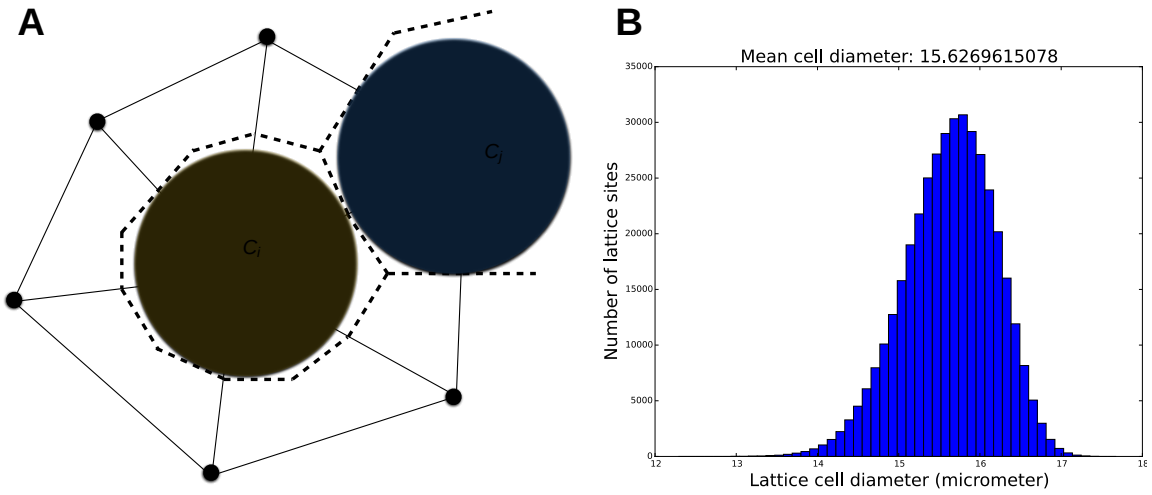
The individual agents - colored coded cells

Each individual cell is modeled as an agent with the following properties:

1. Color (a label used to track the progeny)
2. Mean cell division time, μ_p.
3. Variance in cell division time, σ_{p}^2.
4. Mean migration time, μ_m.
5. Variance in migration time, σ_m^2.

In what follows, we assume that each cell have the same value of the above parameters, i.e. we account only for the intrinsic randomness caused by stochasticity in division and migration times, not the extrinsic randomness that would be caused by different cell lineages or different individuals having different characteristic values of the cell properties, something that would add another source of variability to the overall system. Also, we assume that the distributions of the cell properties do not change in time, i.e. the cell division time and cell migration distributions are constant over the course of a simulation.

Each cell is assumed to occupy one voxel of a tessellation of 3D space, and the mesh resolution is chosen such that the average voxel size matches the desired cell size. In practice, the cell size has to be chosen large enough to obtain a manageable mesh size. The model does not explicitly make use the exact shape and volume of the voxels. Rather, the underlying computational grid is used to define the neighborhood of each cell, via the adjacency information of the mesh. This is illustrated in Computational Model Schematic Fig. 1A. Individual cells are thought to occupy the dual elements (dashed lines) of a unstructured triangular (2D) or tetrahedral primal mesh (3D) (solid lines). When visualized, we draw them as circles or spheres centered on the vertices of the primal mesh, with radius such that the volume corresponds to the dual element. On the unstructured mesh, there will be a size distribution for the mesh elements, i.e. there will be a small variation in the size associated with each lattice site (B). The mesh resolution is chose such that the mean cell diameter is close to the experimentally observed values, but still such that the mesh does not become too fine to permit simulations.



Computational Model Schematic 1. Individual cells are modeled by a number of properties such as their color and distributions for cell division and migration waiting times. The positions of cells in space are tracked on an underlying unstructured lattice, or grid (A). The edges in the primary mesh (solid lines) connect vertices (black dots). A biological cell is modeled by the volume made up of the dual elements (dashed lines), connecting triangle (2D) or tetrahedral (3D) centers and edge or face centers. For visualization purposes, in 3D space, we plot cell individuals as colored spheres with radius equal to the sphere with equal volume as the dual element. Size distribution for the mesh elements for the geometry and mesh used in the simulations in the main paper (B).

Cell division

In our model, the time until a cell, or individual, divides is a random variable. The dividing cell (referred to as the mother cell) create a daughter cell after a normally distributed waiting time τ_D

$$\tau_p \sim N(\mu_p, \sigma_p^2)$$

where the mean division time μ_p and the variance σ_p^2 are parameters of the model to be supplied as input to the simulation. As a measure of the degree of variability in cell division times, we use the standard deviation over the mean

$$f = \frac{\sigma}{m}$$

After division, the daughter cell needs to be deposited on the grid. The division direction, or receiving voxel, is sampled according to a discrete distribution. In the simplest case, all directions of division are equally probable and the direction distribution is uniform. In the general case, weights are assigned according to an external, deterministic gradient.

Cell division direction

The division direction is a random variable. Without a polarizing field, each possible division direction is equally probable. With a polarizing field, the division probability is biased by the gradient. The weights for sampling the division direction when cell C_i divides are taken to be

$$w_{ij} = \left(\frac{d_{ij}}{\max_j(d_{ij})} \right)^b$$

$$d_{ij} = \frac{g(x_j) - g(x_i)}{h_{ij}}$$

The parameter $b \geq 0$ dictates how perfectly the cells become polarized by the concentration profile $g(x)$. A value $b = 0$ leads to equal probabilities for all directions, and very large value of b means that the division direction will always be in the direction of the maximal value of the gradient (the division direction becomes deterministic in the direction of the maximal gradient in the limit $b \to \infty$. Values in between the extremes describe an increasing precision in polarization axis alignment with the gradient field.

Cell pushing

Unlike related individual based models of e.g. solid tumor growth (Poleszczuk and Underling, 2014), we do not assume that cells can only divide into empty lattice sites or that cells become quiescent if they are completely surrounded by cells. Instead, a dividing cell's progeny will push neighboring cells to make room for it. If the receiving lattice site is empty, the daughter cell is simply deposited there. If it is occupied, that cell is displaced to one of its neighbors as illustrated in Computational Model Schematic 2. The probability for the displaced cell to move to a given neighboring grid point depends on the direction of pushing. Let $\boldsymbol{e}_{md}$ be the vector along the edge connecting the mother cell C_m and the daughter cell C_d, pointing towards the daughter cell. Let $\boldsymbol{e}_{\boldsymbol{dk}}$ be the unit vector along the edge connecting the daughter cell C_d and one of its neighbors, C_k. The weight for moving the displaced cell to the neighbor with index \$n\$ is given by

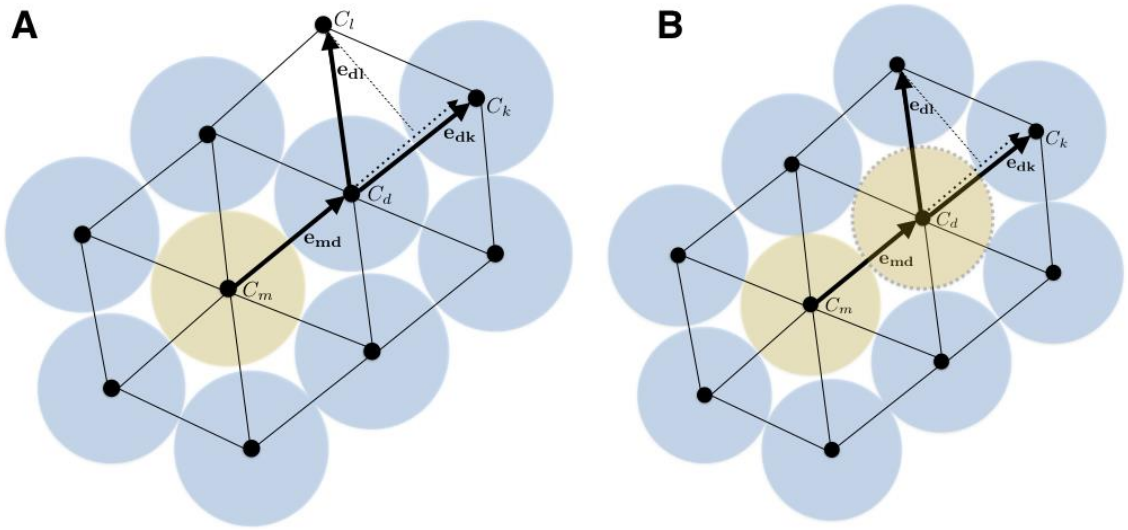
$$w_k = \max(s_k(1 - cI_n), 0)$$

where

$$s_k = \boldsymbol{e}_{\boldsymbol{md}} \cdot \boldsymbol{e}_{\boldsymbol{dk}}$$

That is, to account for the directionality implied by the originally sampled division direction $\boldsymbol{e}_{\boldsymbol{md}}$, the probability of the pushed cell to move to an adjacent lattice site C_k is proportional to

the scalar projection of the vector connecting the pushed cell and the vertex at index k onto the direction vector of the push. $c$is a penalty parameter in the interval $[0,1]$ that dictates the degree of resistance in pushing the cell into an already occupied site. If $c = 1$, it is not possible to push a cell into an occupied site, and if $c = 0$ there is no resistance what so ever and only the direction of the push affects the displacement direction. Any value in between is a tradeoff between the two extremes. In Computational Model Schematic 2, for example, there is a high probability to push C_j to C_k, due to $\boldsymbol{e_{dk}}$ being almost parallel to $\boldsymbol{e_{md}}$(highs_k), but depending on the value of c, the site C_k may be sampled instead since that site is empty, even if the directionality contributions is smaller ($s_l < s_k$). Once a neighbor has been displaced, the pushed cell moves into that lattice site and the daughter cell gets deposited on the now free lattice site of the displaced cell. The displaced cell, may in turn then go on to displace additional cells and this procedure is repeated until a cell gets pushed into an unoccupied site.



Computational Model Schematic 2. When a cell divides (yellow cell), the daughter cell will push surrounding cells (blue) to make room for the progeny (A). The direction of pushing (and what cell to push) is determined by a combination of the directivity of the original division or pushing direction and of a penalty for pushing an occupied site. The penalty is governed by a parameter c that dictates how much to favor pushing into a free lattice site. In the figure, pushing the blue cell at location C_d to C_l will be favored over C_k based on the penalty if $c > 0$, but the site C_k if favored based on the directionality of the push. If C_k is selected by the probabilistic algorithm, it will in turn complete a pushing event, leading to a pushing chain that continues until a cell is pushed into an empty site. After such a pushing chain has been completed, the site sampled for the daughter cell will be free, and the newly created daughter cell can be inserted (B).

This simple model for pushing gives us a practical way to model the insertion of cells without resorting to quiescent cells in the lattice model, or going to a detailed continuum description with plastic cells.

Cell migration

Cells can migrate into a neighboring, receiving, lattice site if it is occupied by another cell or if it is empty. In the case of an empty receiving lattice site, the cell simply changes location, leaving its previous lattice site unoccupied. In the case of an occupied receiving site, the migrating cell and the cell in the receiving voxel swap places. The time to migration is modeled as a stochastic variable with a normal distribution

$$\tau_m = N(\mu_m, \sigma_m^2)$$

The direction of a migration event is chosen uniformly amongst the neighboring lattice sites.

Simulation procedure

A kinetic Monte Carlo algorithm implemented in Python using a priority queue following standard methodology for event driven simulation simulates the system. To initialize the simulation, the time until the next occurrence for each possible event is sampled for all the active individuals. Then, the event with the shortest waiting time is selected, and the corresponding cell updated. In the case of a division event, a new division time is sampled for the mother cell and in the case of migration; a new migration time is sampled for the migrating cell. This procedure is repeated until the simulation has reached its final time point, or until a daughter cell cannot be inserted in the lattice after 10.000 attempted pushing events. This number is selected to be high enough for this never to occur for the simulations times and parameters used in this study.

Computing cell-cell distances with the chain-reaction approach

To mimic the manual analysis conducted on the experimental data, we computed cell-cell distances for each colored cell (clone) using a chain reaction approach. As a starting cell, we picked the one furthest from the mass center of the clone, i.e. a cell on the periphery of the

clonal envelope. Then, we computed the distance to that cell's nearest neighbor. Then that cell was removed from the stack, and the same procedure was repeated for the selected nearest neighbor. This was repeated for 30 cells in the clone (as for the experimental data).