# Robust cell tracking in epithelial tissues through identification of maximum common subgraphs

Jochen Kursawe[1,*], Rémi Bardenet[2], Jeremiah J. Zartman[3], Ruth E. Baker[1], Alexander G. Fletcher[4,5,*]

[1] Mathematical Institute, University of Oxford, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, UK

[2] CNRS & CRIStAL, Université de Lille, 59651 Villeneuve d'Ascq, France

[3] Department of Chemical and Biomolecular Engineering, University of Notre Dame, 182 Fitzpatrick Hall, Notre Dame, IN 46556, USA

[4] School of Mathematics and Statistics, University of Sheffield, Hicks Building, Hounsfield Road, Sheffield, S3 7RH, UK

[5] Bateson Centre, University of Sheffield, Sheffield, S10 2TN, UK

* E-mail: kursawe@maths.ox.ac.uk, a.g.fletcher@sheffield.ac.uk

**Data accessibility:** The datasets and code supporting this article are publicly available under `https://github.com/kursawe/MCSTracker`.

**Competing interests:** We have no competing interests.

**Authors' contributions:** JK, RB, JJZ, REB, and AGF conceived the study and contributed to the manuscript. JK developed the algorithm and carried out the data segmentation and performance analysis. JK, REB, and AGF drafted the manuscript. JJZ provided microscopy data. All authors gave final approval for publication.

**Keywords:** Cell tracking, planar graphs, maximum common subgraph, epithelial sheets

# Abstract

Tracking of cells in live-imaging microscopy videos of epithelial sheets is a powerful tool for investigating fundamental processes in embryonic development. Observing the growth, proliferation, intercalation, and apoptosis of individual cells helps us understand how global morphogenetic processes, such as tissue invagination or extension, are locally regulated and controlled. Accurate cell tracking requires correctly resolving cells moving in and out of field of view between frames, cell neighbour exchanges, cell removal and cell division events. Here, we present a novel algorithm for epithelial cell tracking. The algorithm exploits the graph-theoretic concept of a ‘maximum common subgraph’ to track cells between successive frames of a video. It does not require the adjustment of tissue-specific parameters, and scales in polyonomial time with tissue size. The algorithm does not rely on precise positional information and thus permits large cell movements between frames, enabling cell tracking in data sets acquired at low temporal resolution due to experimental constraints such as photoxicity.

# 1 Introduction

Live-imaging microscopy is a powerful, and increasingly quantitative, tool for gaining insight into fundamental processes during embryonic development [1–3]. Quantitative information on cell growth, proliferation, death, shape changes and movement extracted from live-imaging reveals how such processes are regulated to give correct tissue-level behaviour. This approach has been particularly successful in characterising the growth and patterning of embryonic epithelial tissues in a number of model organisms [4–9].

A common experimental technique for visualising cell shapes in an epithelial sheet is to fluorescently tag a binding molecule, such as E-cadherin (figure 1A). The analysis of time-lapse microscopy data obtained from such tissues is extremely challenging [2, 3], especially in cases of imaging data of rapidly evolving tissues, and when limitations of, for example, microscope speed, imaging resolution or phototoxicity inhibit the creation of datasets with high temporal and spatial resolution.

The analysis of time-lapse microscopy data comprises two major steps: segmentation and tracking (registration). Segmentation must be performed for each frame of a video and involves the identification of objects and landmarks, such as cell shapes (figure 1B). Automated segmentation is hindered by various factors such as noise in fluorescent signals, uneven illumination of the sample, or overlapping cells in a two-dimensional projection. Often, manual correction is necessary to address over-segmentation, where too many cells are detected, or under-segmentation, where too few cells are detected [10–12]. Tracking involves the association of segmented cells across video frames (figure 1C) and requires resolving cellular movement, cell division, cell death, and cells entering and leaving the field of view [12].

Numerous algorithms are available for the segmentation and tracking of cellular-resolution microscopy data [10, 11, 13]. Common methods for cell tracking utilize optimization techniques to minimise differences in cellular properties between two frames [11, 14–17]. The min-cost max-flow algorithm [14] uses linear integer programming to minimise differences in cell areas, perimeters, orientations, and locations between frames, whereas multiple-parameter tracking [15] employs global optimization to minimize differences in cell shapes as well as locations. In contrast, multitemporal association tracking [16, 17] minimises differences in cell locations and sizes by using a probabilistic approach that finds the most likely extension to existing cell

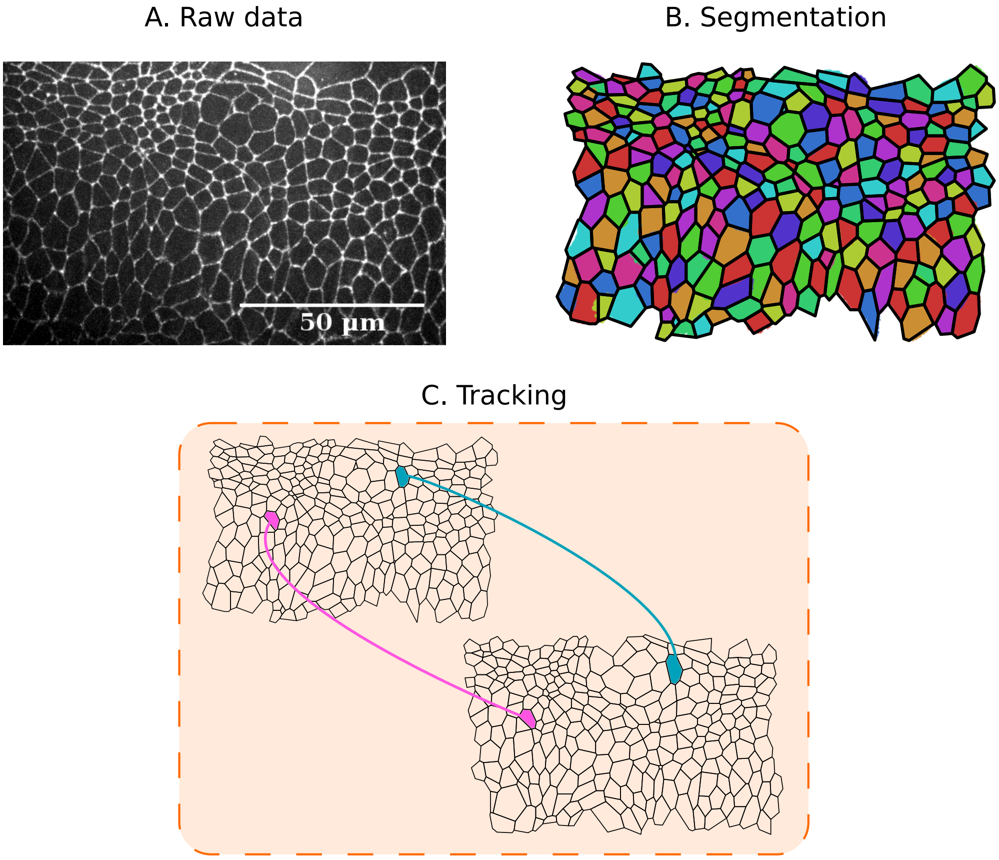

Figure 1: Pipeline for analysing epithelial tissues. (A) Example raw data. Frame of a live-imaging microscopy video of the lateral epidermis of a stage-eleven *Drosophila* embryo, expressing DE-Cadherin::GFP. See Experimental Methods for details. (B) Segmentation of this image, showing cell shapes (coloured regions) and polygonal approximation based on three-cell junctions (black lines). See Methods section for details of segmentation. (C) Cell tracking involves registering individual cells across consecutive segmented images.

trajectories. Chain-graph models [18] minimise differences in cell velocity while overcoming mis-segmentation by verifying that each segmented object continues or begins a cell trajectory in successive frames. Optical flow (‘warping’) between successive frames can be used to guide cell tracking as well as segmentation [19]. It is also possible to combine segmentation and tracking of 2D microscopy videos by interpreting time as a third spatial dimension and employing 3D segmentation techniques [20].

The nearest-neighbour method associates two cells in consecutive frames with each other if their respective centroids have minimal distance within the field of view [10], or if their overlap in pixels within the field of view is maximal [21, 22]. Particle image velocimetry, a technique originally developed to analyse fluid flow [23], has also been employed to track cells in epithelial tissues [24].

Software implementations and computational tools for cell tracking include FARSIGHT [25] (segmentation only), SeedWaterSegmenter [10] (nearest-neighbour tracking), ilastik [18] (chain-graph models), Tufts Tissue Tracker [11] (min-cost max-flow algorithm), Tracking with Gaussian Mixture Models [26] (nearest-neighbour tracking), Packing Analyzer [27] (particle image velocimetry) and EpiTools [13] (nearest-neighbour tracking). These algorithms and software tools primarily rely on there being small differences in cell positions and shapes across consecutive images. Their performance is therefore hindered when analysing data from *in vivo* studies where phototoxicity provides a barrier to high temporal resolution imaging [28–30]. To address this limitation, we propose a novel algorithm for cell tracking that uses only the connectivity of cell apical surfaces (figure 1). By representing the cell sheet as a physical network in which each pair of adjacent cells shares an edge, we show that cells can be tracked between successive frames by finding the *maximum common subgraph* (MCS) of the two networks: the largest network of connected cells that is contained in these two consecutive frames. It is then possible to track any remaining cells based on their adjacency to cells tracked using the MCS. Our algorithm does not require the tuning of parameters to a specific application, and scales in subquadratic time with the number of cells in the sheet, making it amenable to the analysis of large tissues.

We demonstrate here that our algorithm resolves tissue movements, cell neighbour exchanges, cell division, and cell removal (for example, by delamination, extrusion, or death) in a large number of *in silico* data sets, and successfully tracks cells across sample segmented frames from *in vivo* microscopy data. The remainder of the paper is structured as follows. In Section 2 we describe the technical details. In Section 3 we analyse the performance of the algorithm on *in silico* and *in vivo* datasets. Finally, in Section 4 we discuss future extensions and potential applications.

## 2 Methods

We begin with a conceptual overview of our cell tracking algorithm; a detailed description of each step of the algorithm is provided in the section ‘2.1 Mathematical formulation’. The input to the algorithm is a set of segmented images obtained from a live-imaging microscopy data set of the apical surface of an epithelial cell sheet. For each image, the segmentation is assumed to have correctly identified which cells are adjacent and the locations of junctions where three or more cells meet. This information is used to generate a polygonal approximation to the cell tessellation (figure 1B-C). The statistics of polygonal approximations are commonly used to characterise and explore morphological processes in epithelial tissues [11, 31–34].

Our algorithm tracks cells between each pair of consecutive images in three steps (figure 2). First, we use a MCS-approach [35, 36] to generate an initial bijection between the two images that includes every cell whose connections to its neighbours do not change between images, e.g. due to cell rearrangements (figure 2B). Second, we remove from the bijection any cells that have less than three isolated connections to other cells in the MCS (figure 2B-C), since these cells are likely to have been matched incorrectly. Third, we extend the MCS to track any remaining cells that were not included in the bijection and we identify cell division and ‘removal’ (delamination, extrusion or death) events (figure 2D) through characteristic changes to the local cell network under these events.

In the first of the three steps shown in figure 2, the MCS is constructed by iterative extension from an initial seed. The technical details of this iterative extension are described below. Briefly, this initial seed is found by identifying two cells in consecutive images whose neighbourhoods have identical graph structures. The full MCS is then constructed by iteratively adding cells after inspecting MCSs of the cells’ extended neighbourhoods.

## 2.1 Mathematical formulation

**Preliminaries** We begin by introducing the graph theoretic terminology and notation [37] used to describe our algorithm. We consider each pair of successive segmented images as vertex-labelled graphs[1] $G = (V, E)$ and $G' = (V', E')$, respectively. Here and throughout, we use a prime symbol $'$ to refer to the latter of the consecutive images. Each vertex in $G$ or $G'$ corresponds to one cell in the respective segmentation, and two vertices share an edge in the graph if the corresponding cells are adjacent. Throughout, we assume the graphs $G$ and $G'$ to be simple, planar and connected; we emphasise that these graphs represent the dual of the polygonal cell packing (figure 3A). These assumptions are reasonable in the case of simple epithelial cell sheets.

The vertex labelling of $G$ is defined by three functions, $p_G : V \to \mathbb{N}$, $x_G : V \to \mathbb{R}$ and

[1] A *graph* is an ordered pair $G = (V, E)$, where $V \subseteq \mathbb{N}$ and $E \subseteq \{A \subseteq V : |A| = 2\}$. The elements of $V$ and $E$ are called the *vertices* and *edges* of $G$, respectively. Given a graph $G = (V, E)$, a *vertex labelling* is a function of $V$ to a set of labels. With this function, $G$ is called a *vertex-labelled* graph.

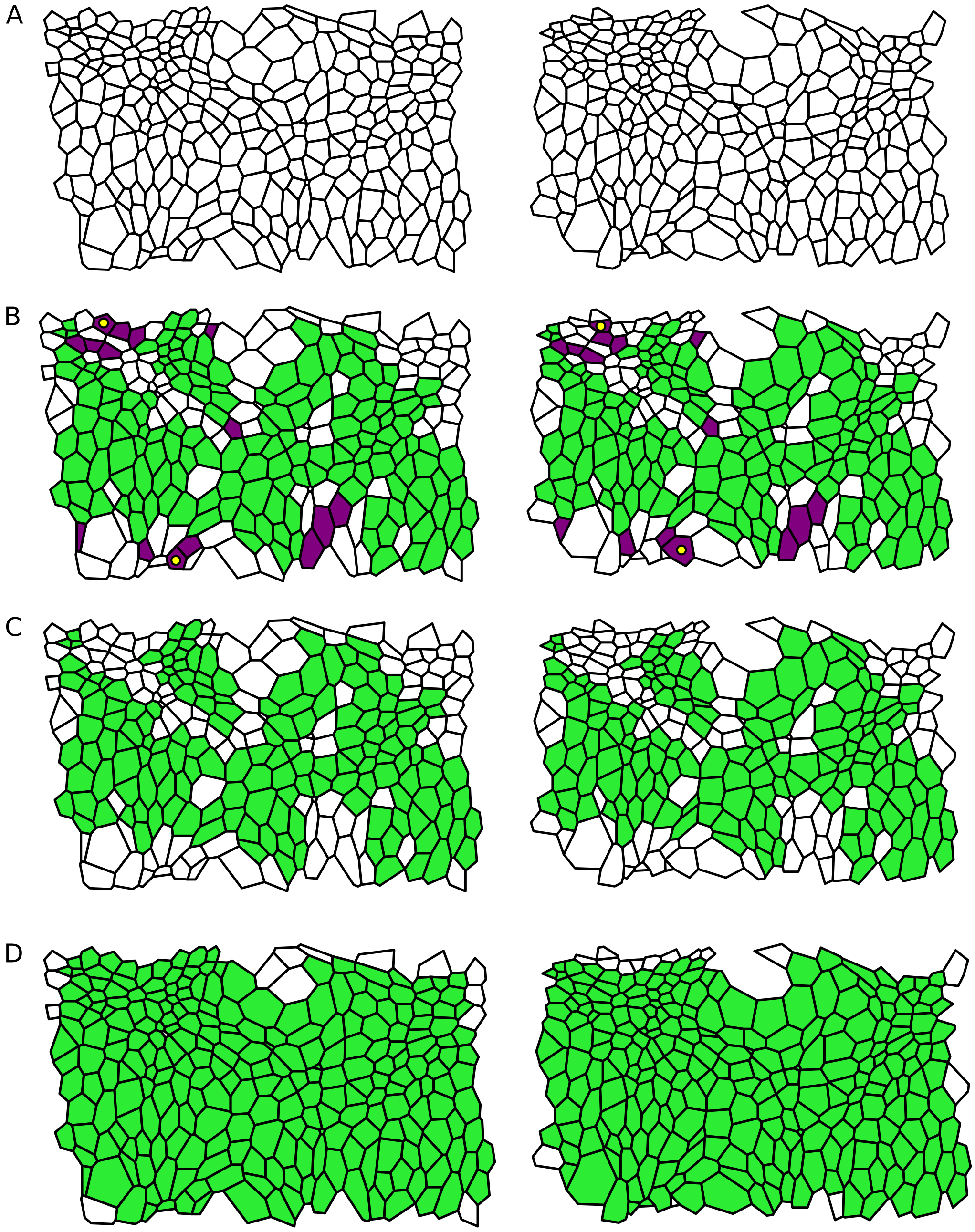

Figure 2: Illustration of our cell tracking algorithm. (A) Two consecutive segmented time-lapse images (left and right columns) of the lateral epidermis of a stage-eleven *Drosophila* embryo, taken five minutes apart. See Experimental Methods for details. There are several cell neighbour exchanges between these images. (B) We first identify a cell mapping between the two graphs based on the conserved MCS. This includes correctly tracked (green/light) cells and weakly connected cells (purple/dark). Here, the conserved MCS incorrectly tracks two cells (yellow/light dots). (C) Weakly connected cells are removed from the conserved MCS to prevent mismatches. (D) An extended tracking mapping is constructed, which includes more cells. See Methods section for details. The remaining white cells enter or leave the frame of view between images and therefore are not tracked.

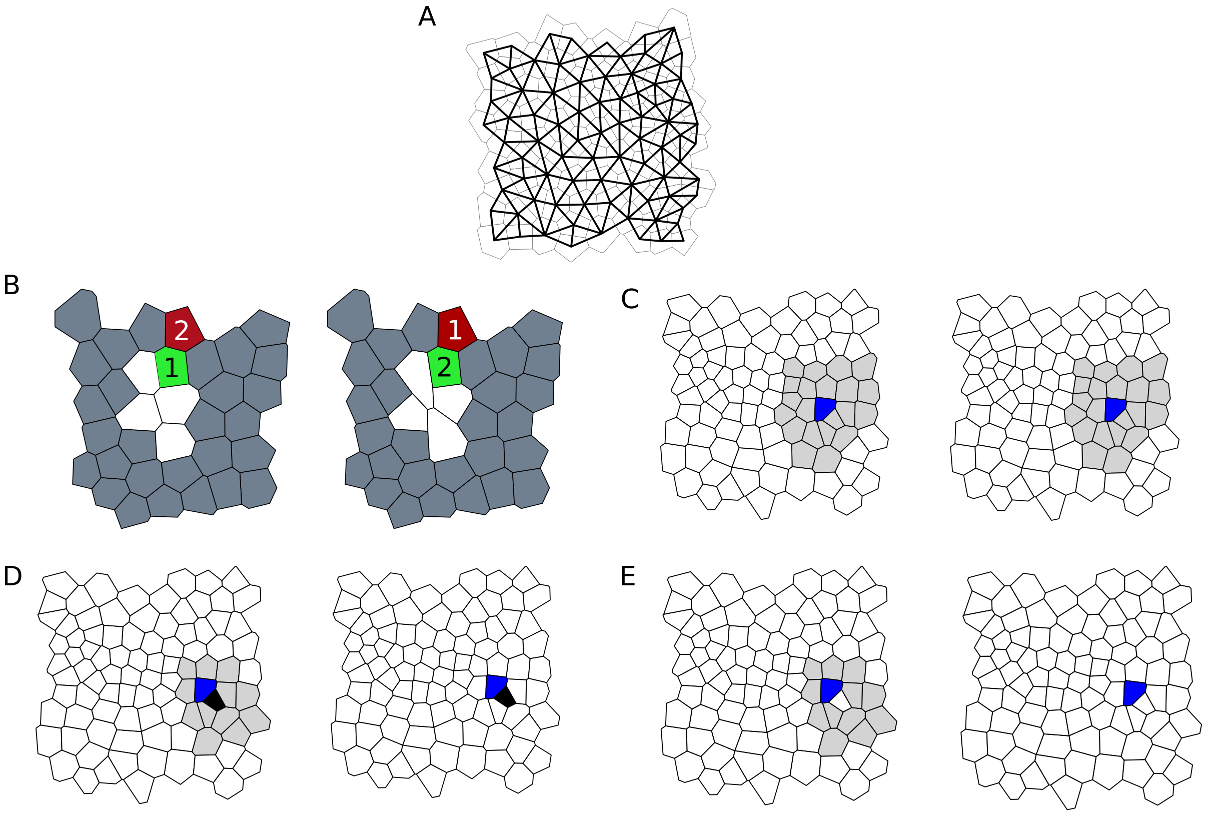

Figure 3: Construction of the MCS. (A) Overlay of a polygonal tessellation (grey) and the corresponding cell network (black). Each cell corresponds to one vertex in the network, and two vertices share an edge if the corresponding cells are adjacent. The network of cells is used by the algorithm to determine the MCS between tessellations corresponding to consecutive time frames in a microscopy video. Note that the network degree of a cell and its polygon number differ at the boundary of the tissue. For example, the highlighted cell has polygon number five and network degree three. (B) The dark grey cells are members of the conserved MCS between the two *in silico* tissues. In this example, two distinct MCSs are possible. Both MCSs include all highlighted grey, green, and red cells. The two MCSs differ in the way the numbered cells are mapped. The first MCS includes the cell pairings as indicated by the green (light) and red (dark) cells. The second MCS includes the pairings as indicated by the numbers one and two. White cells are not members of the MCSs. (C) The algorithm picks a first match of cells for the MCS (blue) if their neighbourhoods form identical networks. The considered neighbourhood includes all neighbours and second nearest neighbours and is shown in grey. (D-E) Additional cells are added to the MCS iteratively by inspecting the MCS between the grey area on the left, and the white area on the right. In (D), where the black cell is paired correctly, the local MCS is larger than in (E), where the selected cell is not considered for mapping. Hence, the pairing of black cells is added to the MCS.

$y_G : V \to \mathbb{R}$. For a vertex $v \in V$, we refer to $p_G(v)$, $x_G(v)$ and $y_G(v)$ as the *polygon number*, *x coordinate* and *y coordinate* of $v$, respectively. For a given vertex, the polygon number is the number of neighbours of the corresponding cell, and the $x$ and $y$ coordinates are defined by the centroid of that cell. An overlay of a polygonal tessellation with the corresponding graph structure is shown in figure 3A.

Let $\phi$ be an isomorphism[2] from $A \subseteq V$ to $B \subseteq V'$ such that for all $v \in A$, we have $p_G(v) = p_{G'}(\phi(v))$ and for all $x, y \in A$, we have $\{x, y\} \in E \Leftrightarrow \{\phi(x), \phi(y)\} \in E'$. We call $\phi$ a *cell mapping* from $G$ to $G'$ and define the *size* of $\phi$ to be $|\phi| = |A|$.

Let $S$ denote the set of cell mappings from subgraphs of $G$ to subgraphs of $G'$. Suppose that $\phi_{MCS} \in S$ has maximum size, i.e. $|\phi_{MCS}| \geq |\phi| \;\; \forall \phi \in S$, and let $V_{MCS} \subseteq V$ denote the domain of $\phi_{MCS}$. We call the subgraph induced[3] by $V_{MCS}$ a *maximum common subgraph* (MCS) of $G$ and $G'$ (this may not be unique). A non-trivial, i.e. non-empty, MCS exists if there are two vertices $v \in V$ and $v' \in V'$ that have the same polygon number, which is always true in our test cases. Our definition of a MCS differs slightly from previous definitions since it requires equivalence of the polygon number in addition to equivalence of edges [35, 38]. Note that the polygon number and degree[4] of a vertex may not coincide for cells at the tissue boundary (figure 3A).

Suppose that $G$ and $G'$ have $k$ MCSs, with associated cell mappings $\phi_1, \ldots, \phi_k$. Let $V_c$ denote the set of vertices in $V$ that are mapped to the same vertex in $V'$ by every cell mapping $\phi_1, \ldots, \phi_k$, and let $\phi_c$ denote the restriction of $\phi_1$ (or, equivalently, any of the cell mappings) to $V_c$. We call $V_c$ the *conserved MCS* of $G$ and $G'$. In contrast to MCSs, conserved MCSs are unique. Examples of MCSs and conserved MCSs are illustrated in figure 3B.

## Construction of the conserved MCS

In general, finding a MCS between two graphs is an NP-hard problem [35]. Here we adapt an efficient MCS detection algorithm [36] by exploiting graph planarity to reduce computational complexity. Instead of exploring all possible combinations of vertex-to-vertex matches [36] we construct the conserved MCS iteratively by finding the MCSs of small subgraphs of $G$ and $G'$. To describe this construction we make use of the following definitions.

For a graph $G = (V, E)$, we define the *extended neighbourhood* of a vertex $v \in V$ to be the set $\Gamma_G^{(2)}(v) = \{w \in V : d_G(v, w) \leq 2\}$, where $d_G$ denotes graph distance[5]. The extended

[2] Graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if there exists a bijection $\phi : V \to V'$ such that, for each $x, y \in V$, we have $\{x, y\} \in E \Leftrightarrow \{\phi(x), \phi(y)\} \in E'$. We say that $\phi$ is an *isomorphism*.

[3] A graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. The subgraph $G'$ of $G$ is *induced* by the vertices $A \subseteq V$ if it contains all edges whose endpoints are both in $A$.

[4] The *degree* of a vertex $v$ of a graph $G = (V, E)$ is the number of incident edges, $\deg_G(v) = |\{w \in V : \{v, w\} \in E\}|$.

[5] The *distance* $d_G(v, w)$ between two vertices $v, w$ of a graph $G$ is the number of edges in a shortest path

neighbourhood contains $v$, all neighbours of $v$, and all second nearest neighbours of $v$. An example of an extended neighbourhood is illustrated in figure 3C as the set of highlighted blue and grey cells.

Let $\rho : A \to B$ be a cell mapping, $v \in V \setminus A$ and $v' \in V' \setminus B$ be vertices in successive graphs, and $S^{\rho}_{LM}$ be the set of cell mappings whose domains lie in $\Gamma^{(2)}_{G}(v)$, whose images lie in $V'$, which map $v$ to $v'$, and which map $v_a$ to $\rho(v_a)$ for all $v_a \in A \cap \Gamma^{(2)}_{G}(v)$. Suppose that $\phi^{\rho}_{LM} \in S^{\rho}_{LM}$ has maximum size, i.e. $|\phi^{\rho}_{LM}| \geq |\phi| \;\; \forall \phi \in S^{\rho}_{LM}$, and let $V_{LM}$ denote the domain of $\phi_{LM}$. We call the subgraph induced by $V^{\rho}_{LM}$ a *local MCS* (LM) of $v$ and $v'$ under $\rho$.

Further, let $S^{\rho}_{RLM}$ denote the set of cell mappings whose domains lie in the extended neighbourhood of $v$ excluding $v$, whose images lie in $V'$, and which map $v_a$ to $\rho(v_a)$ for all $v_a \in A \cap \Gamma^{(2)}_{G}(v)$. Suppose that $\phi^{\rho}_{RLM} \in S^{\rho}_{RLM}$ has maximum size and let $V^{\rho}_{RLM}$ denote the domain of $\phi^{\rho}_{RLM}$. We call the subgraph induced by $V^{\rho}_{RLM}$ a *reduced local MCS* (RLM) of $v$ under $\rho$.

Finally, we say that $v' \in V' \setminus B$ is *mappable* to $v \in V \setminus A$ under $\rho$ if $p_G(v) = p_{G'}(v')$, $d_G(w, v) = 1 \Leftrightarrow d_{G'}(\rho(w), v') = 1$ for all $w \in A$, and if $(x_G(v) - x_{G'}(v'))^2 + (y_G(v) - y_{G'}(v'))^2 < d^2_{\max}$, where throughout this paper we choose the threshold $d_{\max}$ to be ten times the average cell diameter in the tissue (defined as the square root of the average area of the polygonal approximations of the cells in the segmented microscopy image). The threshold $d_{\max}$ is used in our MCS finding algorithm to restrict any possible vertex pairings to those that are in physical proximity. This restriction reduces the size of the search space.

**Initial step** To construct the conserved MCS, we first define a cell mapping $\phi_1$ between single vertices of the consecutive graphs (figure 3C). Formally, we search through vertices in $V$ and $V'$ to find $v_1 \in V$, $v'_1 \in V'$ such that the order[6] of any local MCS of $v_1$ and $v'_1$ under the cell mapping[7] $\phi_0 : \emptyset \to \emptyset$ is equal to $|\Gamma^2_G(v_1)|$ and, for any vertex $v'_2 \in V' \setminus \{v'_1\}$ that is mappable to $v_1$ under $\phi_0$, the order of any local MCS of $v_1$ and $v'_2$ is strictly less than $|\Gamma^2_G(v_1)|$. We then define a first cell mapping $\phi_1 : V_1 \to V'_1$ with $V_1 = \{v_1\}$, $V'_1 = \{v'_1\}$ and a first *set of inspected vertices* $V^{\text{ins}}_1 = \emptyset$. Since we wish to use the MCS to aid our cell tracking, the equivalence of the extended neighbourhoods of $v_1$ and $v'_1$ gives us confidence that the corresponding cells are

connecting them. If no such path exists, then the distance is set equal to $\infty$.

[6] The *order* of $G$ is the number of its vertices, $|V|$.

[7] Here and throughout, $\emptyset$ denotes the empty set.

correctly tracked under $\phi_1$. If we cannot find an initial cell mapping, then the algorithm halts; this means that the cell connectivity changes so quickly that the extended neighbourhood of every cell differs between consecutive images.

**Iterative extension** Our next step is to iteratively construct a cell mapping $\phi_{\text{cell}} : V_{\text{cell}} \rightarrow V'_{\text{cell}}$ for the conserved MCS between $G$ and $G'$, as follows.

For $n = 1, 2, \ldots$, given a cell mapping $\phi_n : V_n \rightarrow V'_n$ and a set of already inspected vertices $V_n^{\text{ins}} \subseteq V$, we determine the set of vertices $S_n \subseteq \Gamma_G(V_n) \setminus V_n^{\text{ins}}$ with at least one mappable vertex in $V' \setminus V'_n$ under $\phi_n$. If there are no such vertices ($S_n = \emptyset$), then we simply define $\phi_{n+1} = \phi_n$, $V_{n+1} = V_n$, $V'_{n+1} = V'_n$, and set $V_{n+1}^{\text{ins}} = \emptyset$. Otherwise, if there are such vertices ($S_n \neq \emptyset$), then we find a vertex $v_{n+1} \in S_n$ with a smallest set of mappable vertices $M'_{n+1} \subseteq V' \setminus V'_n$ under $\phi_n$. We then find all RLMs of $v_{n+1}$ under $\phi_n$ and, for each vertex $v'_m \in M'_{n+1}$, we find all LMs of $v_{n+1}$ and $v'_m$ under $\phi_n$. Next, we find if there is a vertex $v'_{n+1} \in M'_{n+1}$ for which all LMs of $v_{n+1}$ and $v'_{n+1}$ are larger than all LMs of $v_{n+1}$ and $v'_m \in M'_{n+1} \setminus \{v'_{n+1}\}$, and larger than all RLMs of $v_{n+1}$. Finally, we distinguish between the case where $v'_{n+1}$ exists or not. If such a vertex $v'_{n+1}$ exists, then we define a new cell mapping $\phi_{n+1} : V_n \cup \{v_{n+1}\} \rightarrow V'_n \cup \{v'_{n+1}\}$ such that $\phi_{n+1}(v_{n+1}) = v'_{n+1}$ and $\phi_{n+1}(v) = \phi_n(v)\ \forall v \in V_n$, and define a new set of inspected vertices $V_{n+1}^{\text{ins}} = V_n^{\text{ins}}$. If there is no such vertex $v'_{n+1} \in \Gamma_G(V_n) \setminus V_n^{\text{ins}}$, then we construct an extended set of inspected vertices $V_{n+1}^{\text{ins}} = V_n^{\text{ins}} \cup \{v_{n+1}\}$, and set $\phi_{n+1} = \phi_n$, $V_{n+1} = V_n$, and $V'_{n+1} = V'_n$. We then increment $n$ and return to the start of the iteration. Note that at each iteration the algorithm proceeds even if there are no non-trivial LMs or RLMs for a given vertex $v_{n+1}$.

The iteration halts as soon as we encounter $S_n = \emptyset$ for two consecutive values of $n$. We then define $\phi_{\text{cell}} = \phi_n$, $V_{\text{cell}} = V_n$ and $V'_{\text{cell}} = V'_n$. Figure 3D-E illustrates the cells considered when searching for the RLMs and LMs of a given vertex.

## Post-processing

The cell mapping $\phi_{\text{cell}}$ is intended to correctly track as many cells as possible between consecutive images. Nevertheless, it is possible that some members of $V_{\text{cell}}$ may be tracked incorrectly, while the cell mapping may have excluded some vertices in $V$ that could have been tracked correctly. To eliminate tracking errors and track cells that are not included in the conserved MCS, we construct a *tracking mapping*, $\psi_{\text{track}}$, from $\tilde{V}_{\text{track}} \subseteq V$ to $\tilde{V}'_{\text{track}} \subseteq V'$. We call a mapping

$\psi : \tilde{V} \subseteq V \to \tilde{V}' \subseteq V'$ a *tracking mapping* if it is an isomorphism from $\tilde{V}$ to $\tilde{V}'$. In contrast to a cell mapping, a tracking mapping need not preserve polygon numbers or edges between vertices of the subgraphs induced by $\tilde{V}$ and $\tilde{V}'$.

We begin by defining a first tracking mapping $\psi_1 = \phi_{\text{cell}}$ from $\tilde{V}_1 = V_{\text{cell}}$ to $\tilde{V}_1' = V_{\text{cell}}'$. In the following, we describe how we iteratively refine the tracking mapping by first removing vertices from the domain that we suspect to correspond to incorrectly tracked cells (figure 2B-C), and then we add vertices to the domain to track cells that are not members of the MCS (figure 2D).

**Removing weakly connected cells** Let $\psi$ be a tracking mapping from $\tilde{V} \subseteq V$ to $\tilde{V}' \subseteq V'$. We define $v \in \tilde{V}$ to be *weakly connected* with respect to $\psi$ if the set $\Gamma_G(v) \cap \tilde{V}$ contains either: exactly one vertex; or exactly two vertices that are not adjacent. We remove any weakly connected vertices from the tracking mapping since the corresponding cells may have been tracked incorrectly by the MCS (figure 2). To do this, we first find the set of vertices $\tilde{S}_n \subseteq \tilde{V}_1$ that are weakly connected with respect to $\psi_1$. Next, we let $\tilde{V}_2 = \tilde{V}_1 \setminus \tilde{S}_n$, $\tilde{V}_2' = \tilde{V}_1' \setminus \{\psi_1(w) : w \in \tilde{S}_n\}$, and define a new tracking mapping $\psi_2 : \tilde{V}_2 \to \tilde{V}_2'$ to be the restriction of $\psi_1$ to $\tilde{V}_2$. Note that this step accounts for the possibility that $\tilde{S}_n = \emptyset$; in this case, we simply have $\psi_2 = \psi_1$.

**Adding cells that were not tracked by the MCS** We next add cells to the tracking mapping. This is necessary, since any cells that have undergone neighbour exchanges between the consecutive images may have changed their polygon numbers, or their adjacency to each other. This means that their corresponding vertices cannot be members of the conserved MCS, and so regions of cell neighbour exchanges will leave gaps of untracked cells in the MCS (figure 2B-C).

In the following, we iteratively extend the domain of the tracking mapping to include vertices that have neighbours within the domain of the tracking mapping. Possible images of a given vertex can be identified by the aid of the images of the neighbours of the vertex. In this way, we track as many remaining cells as possible based on their neighbour relationships to cells that have been tracked by the conserved MCS. The more mapped neighbours that are preserved between a newly added vertex and its image, the higher our confidence that the corresponding cells are correctly tracked. For this reason, the algorithm starts by requiring that at least $n_p = 4$ previously mapped neighbours are preserved for newly added cells. Once no further cell can be added that fulfils this condition, the algorithm is restarted with requiring $n_p = 3$, and finally with $n_p = 2$.

Formally, we start with a tracking mapping $\psi_n : \tilde{V}_n \to \tilde{V}'_n$ (initially with $n = 2$). We inspect all vertices in $V \setminus \tilde{V}_n$ consecutively. At each step, one such vertex $v$ is considered. Let $T_n(v) = \{\psi_n(w) : w \in \Gamma_G(v) \cap \tilde{V}_n\}$ denote the set of images of all adjacent vertices of $v$ in the domain of the current tracking mapping. If $|T_n(v)| \geq n_p$ (note, that $n_p = 4$ initially), we construct the set of vertices in $V' \setminus \tilde{V}'_n$ that elements of $T_n(v)$ share as neighbours,

$$W_n^{(1)}(v) = \bigcup_{v' \in T_n(v)} \Gamma_{G'}(v') \setminus \tilde{V}'_n. \tag{1}$$

If $W_n^{(1)}(v)$ is empty and $|T_n(v)| \geq n_p + 1$, then we consider reduced sets of images of the form $T_n(v) \setminus \{w'\}$, where one element $w'$ is removed from $T_n(v)$, and we define the set of all shared neighbours of each reduced image set that are not in the image of $\psi_n$:

$$W_n^{(2)}(v) = \bigcup_{w' \in T_n(v)} \left( \bigcap_{v' \in T_n(v) \setminus \{w'\}} \Gamma_{G'}(v') \setminus \tilde{V}'_n \right). \tag{2}$$

By construction, the set $W_n^{(2)}(v)$ contains those vertices in $V' \setminus \tilde{V}'_n$ that are shared neighbours of images of neighbours of $v$, each excluding one such neighbour. We introduce the condition $|T_n(v)| \geq n_p + 1$ above to ensure that the number of preserved neighbours in each reduced image set $T_n(v) \setminus \{w'\}$ is at least $n_p$.

If (i) $W_n^{(1)}(v)$ contains exactly one vertex $v'$, or if $W_n^{(1)}(v) = \emptyset$ and $W_n^{(2)}(v)$ contains exactly one vertex $v'$, and (ii) $v$’ has at most two neighbours in $\tilde{V}'_n$ that are not neighbours of $v$ in $\tilde{V}_n$, then we let $\tilde{V}_{n+1} = \tilde{V}_n \cup \{v\}$, $\tilde{V}'_n = \tilde{V}'_n \cup \{v'\}$ and define a new tracking mapping $\psi_{n+1} : \tilde{V}_{n+1} \to \tilde{V}'_{n+1}$ to be the extension of $\psi_n$ for which $\psi_{n+1}(v) = v'$. Otherwise, if (i) or (ii) is not satisfied, then we leave $\psi_n : \tilde{V}_n \to \tilde{V}'_n$ unchanged. Condition (ii) ensures that cell matches that would add a large number of neighbours to the tracked cell between $G$ and $G'$ are not accepted.

Once $v$ has been inspected, and $\psi_n$ has been extended if possible, a next vertex in $V \setminus \tilde{V}_n$ is chosen and inspected. When all vertices in $V \setminus \tilde{V}_n$ have been inspected, the search is restarted, and all vertices in $V \setminus \tilde{V}_n$ are again consecutively inspected. The search is restarted repeatedly in this manner to ensure that any cells that have gained mapped neighbours during the post-processing step can be inspected for mapping again. We halt our search as soon as $\psi_n$ is not extended between two consecutive restarts. Once the search is halted, we repeat the procedure

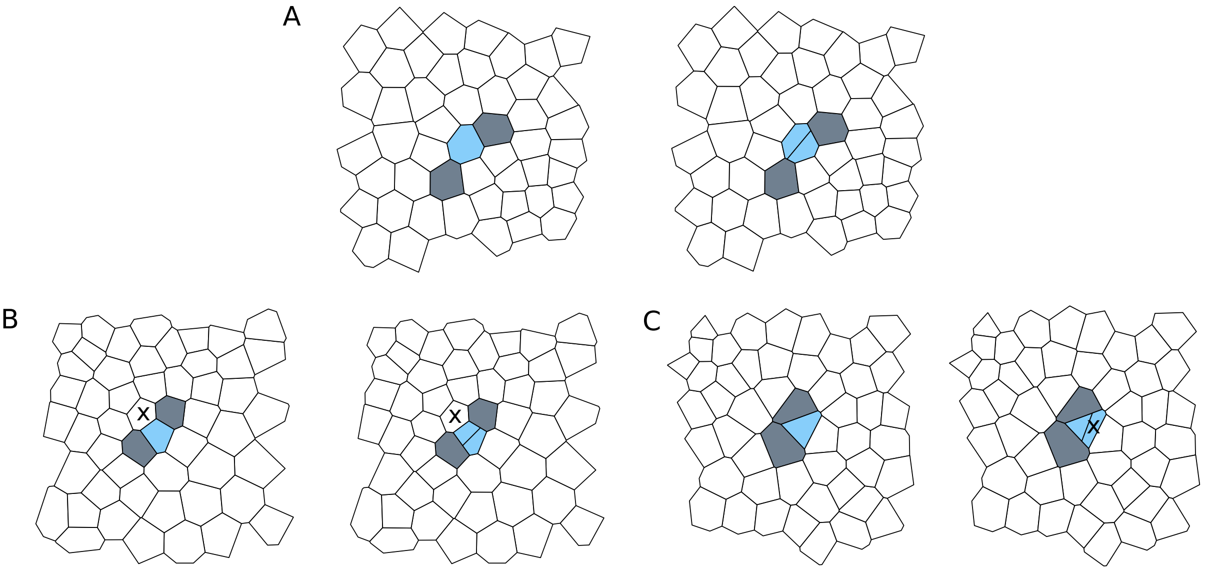

Figure 4: Resolving division events. Dividing cells are coloured blue. (A) Division events are resolved by identifying cells that gain an edge between the time frames (grey cells). The dividing cell and the daughter cells are shared neighbours of the grey cells. (B) When one of the daughter cells is four-sided, two mother cells are possible, the blue marked mother cell, and the cell marked by an 'x'. (C) When one of the daughter cells is three-sided the mother cell can be mistaken as having gained an edge if it is identified with the daughter cell marked by an 'x'. Our algorithm correctly resolves division events such as in (A), (B), and (C).

with $n_p = 3$, and finally with $n_p = 2$.

**Resolving division events** If a cell divides between consecutive frames, then the tracking mapping $\psi_n$ we have constructed thus far may incorrectly identify the mother cell with one of its daughter cells (figure 4). To address this issue, we construct a tracking mapping $\psi_{\text{track}}$ in which incorrectly tracked mother cells are removed. To resolve division events, we first identify *boundary vertices* to be those vertices $v \in V$ whose polygon number and degree differ. This corresponds to cells that are at the physical boundary of the sheet, where polygon number and network degree do not coincide (figure 3A). We then identify all connected sets of vertices $M' \subseteq V' \setminus \tilde{V}'_n$ that satisfy $\Gamma'_G(M') \subseteq \tilde{V}'_n$ and that contain no boundary vertices of $V'$. Each such set $M'$ corresponds to one division event, and in the following we treat each $M'$ individually.

For each $M'$, we define $S_{M,1} = \psi_n^{-1}(\Gamma'_G(M'))$ to be the set of inverse images of the mapped neighbours of $M'$ under $\psi_n$. Next, we identify the set $S_{\text{border}} \subseteq S_{M,1}$ of potential bordering cells of the division, i.e. cells that are adjacent to the division, by finding those vertices $v \in S_{M,1}$

that gain an edge under the tracking mapping $\psi_n$:

$$S_{\text{border}} = \{v \in S_{M,1} : p_{G'}(\psi_n(v)) = p_G(v) + 1\}. \tag{3}$$

We also identify the set $S_{\text{mother}}$ of potential mother cells by finding any shared neighbours of potential bordering cells:

$$S_{\text{mother}} = \bigcap_{v \in S_{\text{border}}} \Gamma_G(v). \tag{4}$$

Based on the sets $S_{\text{border}}$ and $S_{\text{mother}}$ we decide which cells are the mother and daughter cells of the division event, distinguishing between the following cases:

(i) If $S_{\text{mother}}$ contains exactly one vertex, then this is identified as the mother cell of the division, and $M'$ must contain exactly two vertices, which are identified as the daughter cells. In this case, neither the mother nor daughter cells are three- or four-sided.

(ii) If $S_{\text{mother}} = \emptyset$, then one of the daughter cells must be three-sided (figure 4c). In this case, a geometry-inferred selection of mother and daughter cells is required. To this end, we define a set of potential daughter cells

$$S'_{\text{daughter}} = \psi_n(S_{\text{border}}) \bigcup \left( \bigcap_{v' \in \psi_n(S_{\text{border}})} \Gamma_{G'}(v') \right). \tag{5}$$

that contains the images of the potential bordering cells and all shared neighbours of these images in $V'$. Next, we find a definite daughter cell as an element $v' \in S_{\text{daughter}}$ that is three-sided ($p_{G'}(v) = 3$). The geometry-inferred selection of the second daughter cell proceeds as follows. For each $w' \in S'_{\text{daughter}} \setminus \{v'\}$, we construct the *geometrically merged cell* of $v'$ and $w'$ by removing the edge between the polygons that corresponds to $v'$ and $w'$ in the segmentation of the microscopy video frame from which the graph $G'$ was generated, as well as the cell junctions where three or more cells meet at the end of this edge. We then calculate the distance of the centroid of the geometrically merged cell to the centroid of the cell associated with vertex $\psi_n^{-1}(w')$. The vertex $w'$ for which this distance is minimal is identified as the second daughter cell, and the mother cell is identified as its inverse image under $\psi_n$.

(iii) If $S_{\text{mother}}$ contains more than one vertex, then we define a set of potential daughter cells as any shared neighbours of images of the potential bordering cells

$$S'_{\text{daughter}} = \bigcap_{v' \in \psi_n(S_{\text{border}})} \Gamma_{G'}(v'). \tag{6}$$

If $S'_{\text{daughter}}$ contains exactly four vertices, then the mother cell and both daughter cells are four-sided, and the mother cell can be identified as the single vertex in the set $S_{M,2}$, which we define as the set of cells which are shared neighbours of all cells in $S_{M,1}$ (the inverse images of neighbours of the division), and which are not in the domain of $\psi_n$, i.e.

$$S_{M,2} = \bigcap_{v \in S_{M,1}} \Gamma_G(v) \setminus \tilde{V}_n \tag{7}$$

The daughter cells correspond to the only two vertices in $M'$.

If $S'_{\text{daughter}}$ contains exactly three vertices, then one of the daughter cells is four-sided, and we identify this cell as the definite daughter cell of the division $v'$, i.e. we identify $v' \in S'_{\text{daughter}} : p_{G'}(v') = 4$. In this case, geometry-inferred selection of the second daughter cell is required, and we achieve this in a similar way to that described for three-sided daughter cells above. For each cell $w' \in S'_{\text{daughter}} \setminus \{v'\}$, we construct the merged cell of $v'$ and $w'$, and calculate the distance of its centroid to the centroid of $\psi_n^{-1}(w')$. The cell $w' \in S'_{\text{daughter}} \setminus \{v'\}$ for which this distance is smallest is the second daughter cell. Since in this case $S_{mother}$ contains more than one cell, $S'_{\text{daughter}}$ must contain at least three cells[8].

Once each set $M'$ has been inspected and the associated division event has been resolved by identifying the mother and daughter cells, we construct a tracking mapping in which any incorrectly tracked mother cells are removed. To this end, we define the set of all mother cells for which geometry-inferred selection has been used as $S_{\text{geo}}$, and we construct a final tracking mapping $\psi_{\text{track}} : \tilde{V}_n \setminus S_{\text{geo}} \to \tilde{V}'_n \setminus \psi_n(S_{\text{geo}})$ such that $\psi_{\text{track}}(v) = \psi_n(v)\ \forall v \in \tilde{V}_n \setminus \psi_n(S_{\text{geo}})$.

In general, the division resolution step may incorrectly track cells in cases where there is a cell neighbour exchange next to the division, or if there are two adjacent divisions between frames. For example, if each of the bordering cells, i.e. the cells adjacent to the division, were to undergo a neighbour exchange in which they lose an edge between images, then our algorithm

[8] If $S'_{\text{daughter}}$ contains more than four cells, then our algorithm fails; however, this was never encountered in our test cases.

would fail to correctly resolve the division event.

**Resolving remaining events** At this stage, the tracking algorithm for the two consecutive time frames is completed, and it is straightforward to identify cell neighbour exchanges by finding any cells that have changed their polygon number from one frame to the next. Cell removal events correspond to any vertices $v \in V$ that are not in the domain of $\psi_{\text{track}}$, and for which $\Gamma_G(v) \subseteq V_{\text{track}}$, and that do not correspond to mother cells of a division event.

## Computational implementation

We use Krissinel's MCS finding algorithm [36] to find all RLMs and LMs in the above steps. This algorithm will always halt eventually. In particular, since the domains on which the RLMs and LMs are calculated only contain extended neighbourhoods of individual cells, the MCS finding does not pose computational barriers. We adapt the procedure for MCS finding proposed in [36] in two ways: (i) whenever a next vertex is considered for mapping, we pick a vertex that is adjacent to already mapped cells, hence the adapted algorithm only finds connected subgraphs; (ii) since the RLMs and LMs are small, we do not implement subgraph-size dependent conditions to interrupt the search early.

When finding the initial mapping, for any two possible matches LMs are first calculated by considering nearest neighbours only rather than extended neighbourhoods. Once the neighbourhoods[9] of two matching vertices are found to be isomorphic, the extended neighbourhood is considered. This step reduces the time that is needed to find the initial match.

In the computational implementation of the tracking algorithm we use a further vertex-label $c_G : V \to \mathbb{N}$, which we call the *cell identifier*. In practice, integer identifiers for a given vertex $v$ arise naturally in the segmentation step. Cell identifiers allow us to easily identify vertices and relate them to a cell in a given image independent of how they are stored in the graph structure.

The code used in this article is publicly available under the 3-clause BSD license as the MCSTracker project (`https://github.com/kursawe/MCSTracker`). The project is implemented in pure-Python, employs unit-testing [39] and is fully documented. Graphs in our code are represented using the NetworkX package in Python [40].

[9] The set of adjacent vertices, $\Gamma_G(v) = \{w \in V : \{v, w\} \in E\}$ is called the *neighbourhood* of $v$, so the degree of $v$ is $|\Gamma_G(v)|$. We define the neighbourhood of a subset $V' \subseteq V$ to be $\Gamma_G(V') = \{w \in V \backslash V' : \exists v \in V' \text{ with } d(w, v) = 1\}$.

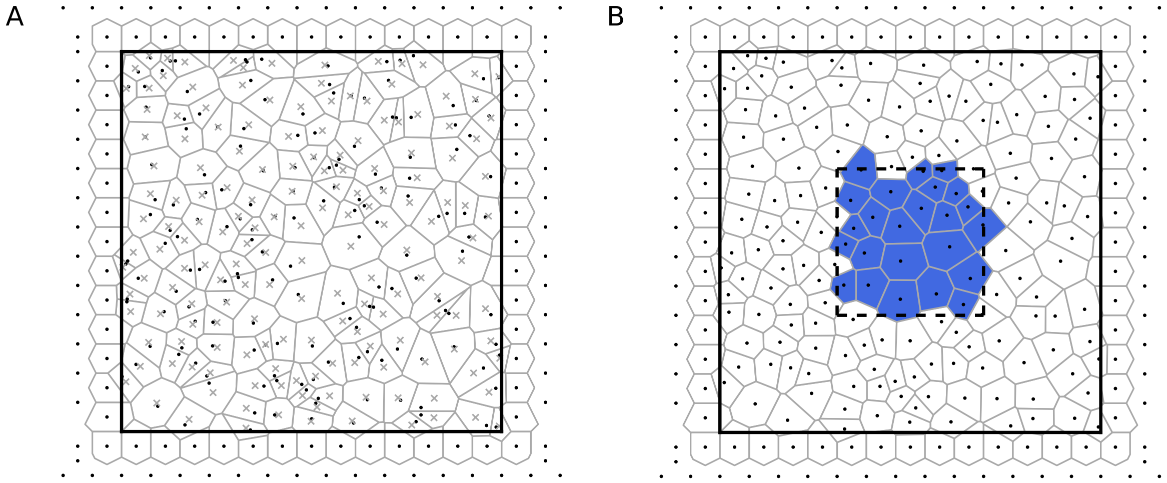

Figure 5: Generation of *in silico* data. (A) Random seeds (black dots) are placed inside a domain $\Omega$, the border of which is shown using a black line. Additional, evenly spaced seeds are placed outside $\Omega$. The Voronoi tessellation of all seeds is shown in grey, excluding Voronoi regions corresponding to the outermost row of seeds, since these are large or unbounded. The centroids of the Voronoi regions differ from the seeds, and are shown as grey crosses. (B) The centroids of the Voronoi regions in (A) are used as seeds for a new Voronoi tessellation, for which evenly spaced seeds are again added outside the domain $\Omega$. Voronoi regions whose centroids lie within a window (dashed black line) at the centre of the domain are collected to form the *in silico* tissue (blue). In this figure, one Lloyd's relaxation step ($n_L = 1$) is shown. Throughout this study, we generate *in silico* tissues using $n_L = 4$ Lloyd's relaxation steps.

## Generation of *in silico* data sets

To test the algorithm, we generate *in silico* data sets that include examples of cell divisions, removals and neighbour exchanges, as well as tissue movement. These data sets are generated using Voronoi tessellations modified using Lloyd's relaxation, which resemble cell packings in a variety of epithelial tissues [33, 41].

To generate polygonal patterns of size $m \times n$, where $m$ and $n$ are natural numbers, $(m + g) \times (n + g)$ Voronoi seeds are uniformly random distributed in a 2D domain $\Omega$ of width $m + g$ and height $n + g$ (figure 5A). Here, $g$ is the size of a boundary region that is introduced to reduce the impact of the Voronoi boundary on the patterns. The domain $\Omega$ is surrounded by two rows of evenly spaced additional seeds on each side. The inner row has a distance of 0.5 to $\Omega$, and the seed-spacing is 1.0. The outer row has a distance of 1.5 to $\Omega$, and the seeds are shifted parallel to the first row by a distance of 0.5. The Voronoi tessellation of all these seeds is then constructed.

In each Lloyd's relaxation step, the polygons (or infinitely large areas) corresponding to the regularly spaced seeds outside $\Omega$ are removed from the tessellation. Next, the centroid of each remaining polygon is calculated and registered as a new seed. Further seeds are added that again correspond to two rows of evenly spaced seeds outside $\Omega$. A new Voronoi tessellation is then constructed (figure 5B). This procedure is repeated for $L$ relaxation steps, after which all generated polygons are discarded except those whose centroid lies within an area occupying $n \times m$ area units in the centre of $\Omega$ (figure 5B).

The polygonal tessellations have approximately $m \times n$ polygons of average area 1.0. During the generation of the tessellations, evenly spaced seeds outside $\Omega$ are added to prevent the occurrence of infinitely large polygons inside $\Omega$. The boundary of size $g$ is added in between the generated tessellation and the evenly spaced seeds in order to reduce the effect of the evenly spaced boundary seeds on the tessellation. Throughout this study, we use $g = 8$ and $n_L = 4$, resulting in cell packings similar to those observed in the *Drosophila* wing imaginal disc [33]. We provide further details of how tissue rearrangements are implemented in the Results section.

## Experimental methods

Live-imaging of cell proliferation was performed in stage-eleven *Drosophila* embryos expressing a tagged version of DE-Cadherin (DE-Cadherin::GFP) using a spinning disc confocal microscope, as described in [42]. For the embryo setup, a modified version of the standard live-imaging protocol was used [43].

**Data segmentation** Microscopy images were segmented manually using SeedWaterSegmenter [10]. Each segmentation was saved as a 16-bit grayscale image where pixels belonging to different cells have different integer values. Polygonal tessellations for the tracking algorithm were generated from the segmented image in two steps. First, all junctions between three or more cells were identified as points where pixels of three or more different cells meet, and second, vertices were assigned to cells. Finally, edges shorter than two pixels (0.5 µm) were removed and replaced by a single vertex at the midpoint of the edge.

# 3 Results

***In silico* testing of the algorithm.** To assess the performance of the algorithm, we begin by applying it to *in silico* data sets that include cell neighbour exchanges, tissue movement, cell removal and cell division, respectively. In each case, we compare the outcome of the tracking algorithm to the ground truth.

We begin by assessing the ability of the algorithm to resolve permutations in otherwise identical tissues (figure 6A). In this test, a random tessellation of size nine by nine cells is created as described in the Methods section, and integer identifiers $c_i$ are assigned to each cell. Next, an identical copy of the tissue is created in which the integer identifiers are randomly shuffled. A ground truth mapping from the first to the second integer identifiers is generated. Next, the algorithm is applied. Upon conducting 100 such tests, all identified cell-to-cell mappings are matched correctly, as compared to the ground truth. In rare examples, isolated cells at the boundary of the tissue are are not tracked. In these examples, either a single cell has only one adjacent cell in the tissue, or two cells of identical polygon number are adjacent and share exactly one neighbour. Neither the MCS detection algorithm, nor the post-processing algorithm are able to resolve such mappings, which involve fewer than four cells in each dataset (fewer than five percent of the tissue).

We design four further tests of tissue rearrangements (figure 6B-E). The first test comprises tissue movements between images (figure 6B). In this test, a tissue of size fifteen by eight cells is generated as described in the Methods section. Two smaller tissues of width seven are cut out of this tissue, which each cover the full height of the tissue, and which are horizontally translated relative to each other by a distance of two cell lengths. The position of each three-cell junction in both tissues is shifted such that the $x$-coordinate of the left-most junction in each tissue is 0.

The second test (figure 6C) generates cell neighbour exchanges, also called T1 transitions [44, 45]. In our implementation of T1 transitions, an edge shared by two cells is replaced by a new perpendicular edge (of length $l_{T1} = 0.2$ units) such that the local cell connectivity changes (figure 2B). We create two identical copies of a tissue of size nine by nine cells. In the second copy, a T1 transition is performed on an edge in centre of the tissue.

The third test involves cell removal (figure 6D). In this test, we first generate two identical copies of a tissue of size nine by nine cells. In the second copy, we replace the central cell by a

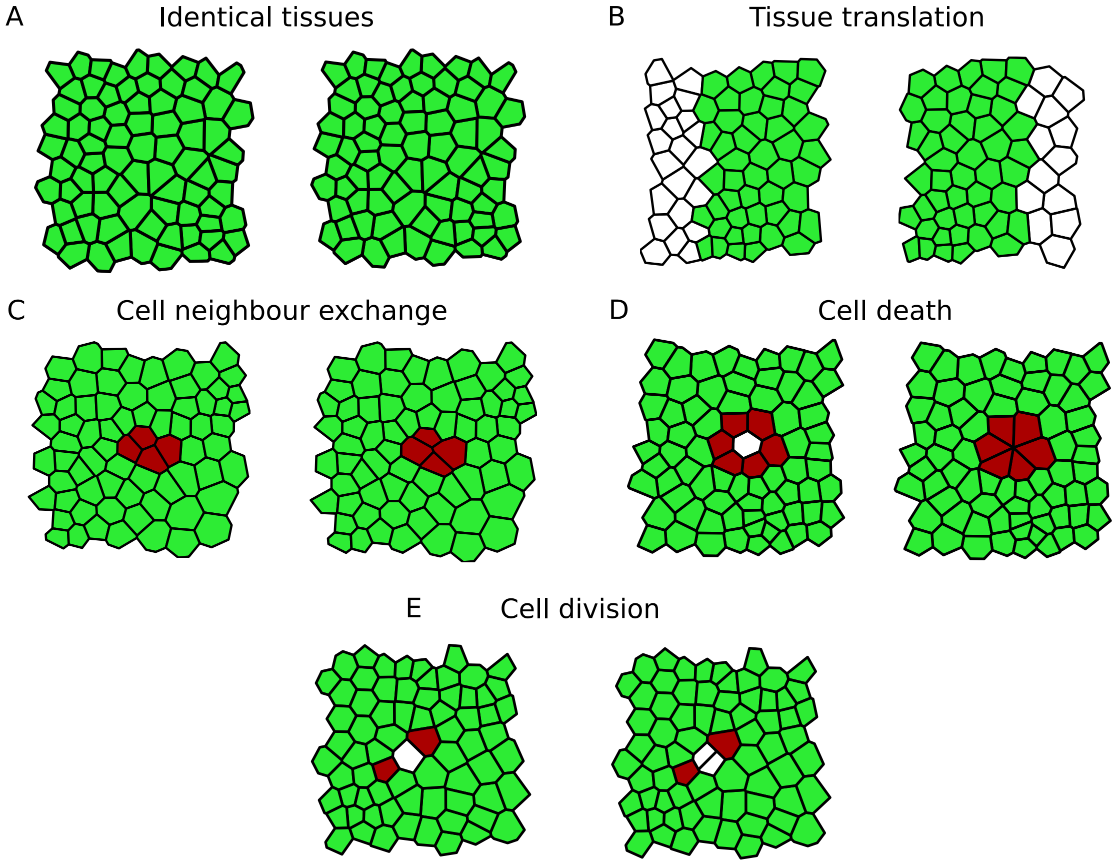

Figure 6: Examples of *in silico* test cases. In each image, cells identified by the MCS algorithm are highlighted in green (light), whereas cells that have been filled in by the post-processing steps are highlighted in red (dark). The algorithm tracks cells between identical tissues (A), in tissues undergoing translation (B), cell neighbour exchange (T1 swap) (C), cell removal (D) and cell division (E).

vertex shared by its neighbouring cells. This rearrangement is similar to so-called T2 transitions in the foams literature [44]. The final test involves cell divisions (figure 6E). Here, we once again create two identical copies of size nine by nine cells. In the second copy, a cell in the centre of the tissue is divided by introducing a straight line in a random direction through centroid of that cell.

For all tests generated in this way, integer cell identifiers in the second tissue are randomly shuffled, and a ground truth is generated. We run 100 realisations of each test case, and compare the tracking outcome to the ground truth. In all cases cells are tracked correctly, with at most three unmatched cells at the boundary of the sheet.

In figure 6, all cells identified after the cleaning step, in which weakly connected cells are removed from the MCS, are coloured green, whereas cells that are identified by the post-processing algorithm are coloured red. Note that the exact number of cells that are identified by the post-processing algorithm varies between individual realisations of the tests. In many cases, the cells identified by the post-processing algorithm include cells that are adjacent to the cells that are undergoing division, removal or neighbour exchange.

We next analyse the extent to which the success of our tracking algorithm depends on the number of Lloyd's relaxation steps, $n_L$, used to generate the *in silico* data sets. To investigate this we iteratively increase $n_L$, and so generate tissues with increasingly homogeneous graph structures, and repeat all tests. We find that the algorithm successfully passes all tests for all values of $n_L$ from 4 up to 14.

**Application of the algorithm to *in vivo* data** Figure 7 shows three sample segmented images of the lateral epidermis of a stage-eleven *Drosophila* embryo, taken five minutes apart, and to which we apply the algorithm. These images comprise 271, 263 and 263 cells, respectively. Between the first and the second images, 247 cells are tracked, whereas 245 cells are tracked between the second and third images. The number of cells that are tracked across all three images is 234. The centroids of cells of previous images are plotted on top of the tracking, showing that the tracking algorithm successfully tracks cells in situations where it is difficult to match cells between images based on the centroid positions alone. Cells that include only their corresponding centroid from the previous image are coloured in green, whereas cells that do not include their corresponding centroid from the previous image, and cells that include multiple centroids from the previous image, are coloured in purple.

On average, cell centroids move 0.75 cell lengths between the first and second images, with a maximal displacement of 1.17 cell lengths. Between the first and second images 36 cells undergo a net gain in edges, whereas 20 cells have net loss of edges. In total, four cell deaths and no cell divisions are observed across all three data images, and none of the cells are tracked incorrectly.

**Calculation times** To analyse the scaling of the calculation times with tissue size we repeat the permutation test with tissues of square dimension of varying size on a desktop computer with an Intel i5-6500T CPU (2.5GHz) and 8GB R. We find that the calculation times scale subquadratically with cell number (figure 8).

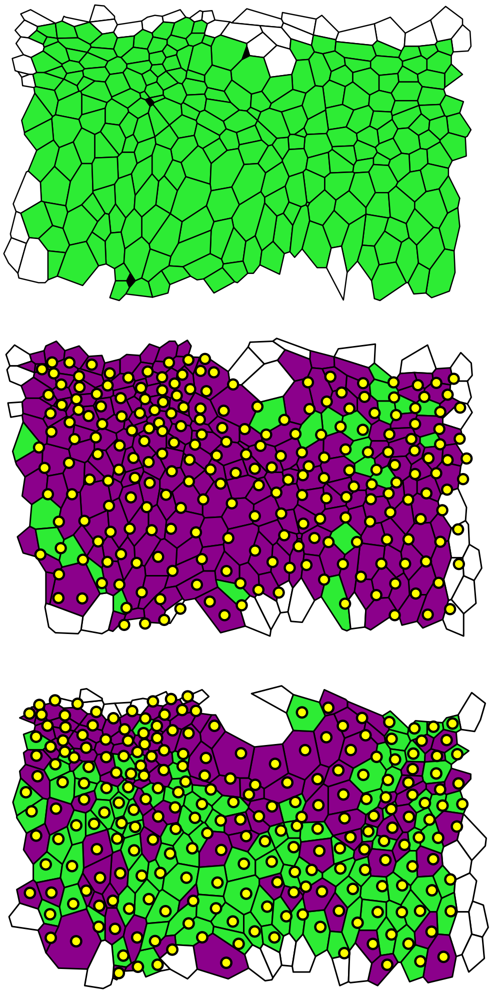

Figure 7: Three segmented data frames. Cells that are tracked across all frames are coloured green or purple, and cells that leave or enter the tissue at the boundary are white. Dying cells are black. The centroids of tracked cells of the respective previous frames are included as yellow dots, and cells that contain only their centroid from the previous frame are coloured green, whereas cells that do not contain their centroid from the previous frame, and cells that contain multiple centroids, are coloured purple. Together, the centroids and the colouring illustrate that it is challenging to track cells between the data frames using solely centroid positions.

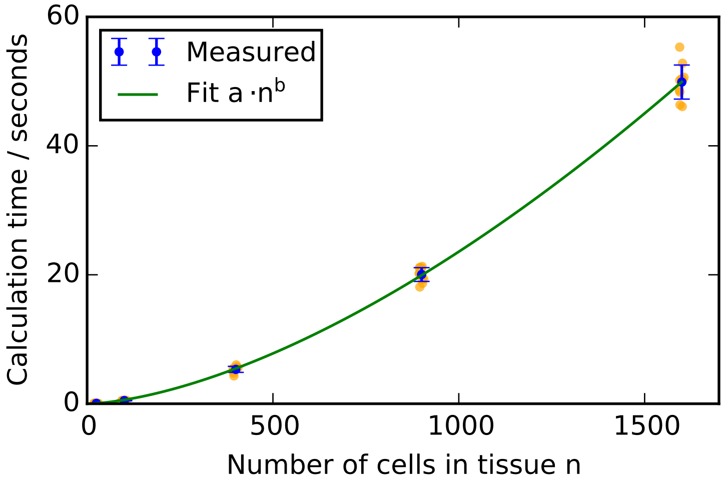

Figure 8: Scaling of the calculation times with tissue size. Virtual tissues of square dimension of varying sizes were generated and the calculation times of the algorithm under the permutation test in figure 6A recorded. Orange dots represent calculation times for individual realisations of the test and error bars denote the standard deviation. The exponent $b$ of the polynomial fit is 1.6. The calculation times were measured on a desktop computer with an Intel i5-6500T CPU (2.5GHz) and 8GB R.

The calculation times for the experimental images analysed in figure 7 vary more widely than for the *in silico* data sets. For the tracking between the first and second frames in this figure, the algorithm required 96 seconds to run, whereas between the second and the third frames the algorithm required 15 seconds. This difference appears to arise from differences in the time required to find the first correct mapping. In the first example 154 cells were searched before the first correct mapping was found, whereas in the second example only 12 cells were searched. This means that the number of cells considered when finding the initial mappings depends on the graph structure of the analysed frames and impacts on the calculation time of the algorithm.

**Algorithm performance on rearranging tissues** To assess the performance of the algorithm on rearranging tissues, we applied the algorithm to *in silico* data sets with increasing numbers of cell rearrangements (figure 9). The number of correctly tracked cells decreases as the number cell rearrangements increases. However, the number of incorrectly tracked cells remains low even for large numbers of cell rearrangements.

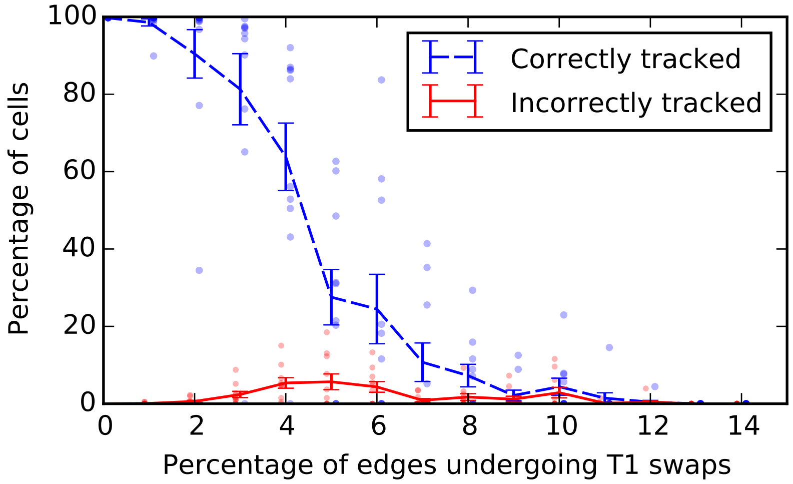

Figure 9: Success rate of the algorithm on *in silico* tissues with increasing amounts of cell rearrangements. Virtual tissues spanning 20 cell lengths in each dimension are generated, and T1 swaps are applied to an increasing proportion of the inner edges of the tissue. For each ratio of T1 swaps, 10 repetitions of the test are run, and the ratio of correctly and incorrectly tracked cells in the tissue is recorded. The dashed blue and solid red lines correspond to mean values of correctly and incorrectly tracked cells, respectively. Error bars denote the standard deviation of the mean, and results of individual runs of the test are represented by dots.

The number of untracked cells increases rapidly as the percentage of edge rearrangements, i.e. the percentage of inner edges in the tissue that are swapped between successive images, increases from five to ten percent. Note that the number of cells involved in these cell rearrangements is larger than five to ten percent, since an individual T1 transition changes the cell neighbour relations of four different cells, and each cell shares multiple inner edges. For example, rearranging five percent of the inner edges of the tissue affects roughly 40 percent of the cells in the tissue, whereas rearranging ten percent of the tissue edges affects up to 70 percent of the cells. The number of (correctly or incorrectly) tracked cells drops to zero if the tissue rearranges so much that the extended neighbourhood of each cell rearranges. In this case a first match cannot be found to initialise the MCS construction algorithm.

## 4 Discussion

Cell tracking in epithelial sheets has the potential to generate a vast amount of quantitative data to inform our understanding of the contributions of different cellular processes to tissue morphogenesis. However, cell tracking is notoriously difficult, especially for the complex morphogenetic processes that occur as embryogenesis proceeds. Here, we present an algorithm based on MCS detection for the tracking of cells in segmented images of epithelial sheets. Our algorithm successfully tracks cells in *in vivo* images of the *Drosophila* embryonic epidermis, as well as in randomly generated *in silico* data sets, without the need for the adjustment of tissue specific parameters, such as weights for individual terms in a global minimisation scheme [14]. The use of *in silico* data to test our algorithm allows us to analyse the performance of our algorithm for a large range of experimentally observed cell rearrangements and tessellations.

Our algorithm is able to track cells that undergo significant movement and neighbour exchanges between frames. For example, we can correctly track cells in tissues where more than 40 percent of the cells rearrange between successive movie frames (figure 9). In addition, even comparably large gaps in the initial MCS can be filled in during the post-processing step (figures 2 and 7). For example, in the first tracking step in figure 1, only 182 of the 246 tracked cells were identified by the MCS algorithm, and it was possible to track the 64 remaining cells during the post-processing step. For comparison, Heller et al [13] report 15 cell rearrangements per 1000 cells per hour at an imaging interval of six minutes for their time-laps microscopy data of *Drosophila* wing imaginal discs. In addition, the experimental data shown in figures 2 and 7, as well as our *in silico* cell removal data sets, contain multicellular rosettes, hence rosettes do not pose a challenge to our algorithm.

Our algorithm is able to correctly track cells in all considered test cases. However, on rare occasions a few cells at the tissue boundary cannot be tracked. It may be possible to adapt the algorithm to track these cells, if this is considered necessary for the application at hand. In the current version of the algorithm, two connections to already tracked cells that are preserved between two time frames are a condition to add a cell-to-cell mapping in the post-processing algorithm. Further analysis of cases where this condition is not fulfilled may reveal ways to relax it.

When generating *in silico* data to test the algorithm, we used Voronoi tessellations in combination with Lloyd's relaxation to generate data that resembles tissues in the *Drosophila* wing imaginal disc [33]. We expect the algorithm to perform less well on tissues whose network structure is near homogeneous. For example, in an epithelial sheet where cells are arranged in a hexagonal fashion, such as the early *Drosophila* embryonic epidermis [46] or the late pupal *Drosophila* wing [47], the local adjacency network of each cell is identical, and hence a network-based tracking algorithm may not be able to distinguish cells. When generating *in silico* tissues, we use four Lloyd's relaxation steps after Voronoi tessellation. With each Lloyd's relaxation step, the homogeneity of the tissue increases. We were able to successfully repeat all *in silico* tests on virtual tissues that were generated using up to $n_L = 14$ Lloyd's relaxation steps. Hence, we expect the algorithm to be suitable for tissues that can be well described with 14 or fewer Lloyd's relaxation steps, such as the chick neural tube embryonic epithelium, or the *Drosophila* eye disc [33].

The algorithm relies on being able to generate polygonal tessellations from segmented video microscopy data. In particular, all *in silico* tests we conducted consider tissues where each cell has at least three neighbours. Conceptually, it would be possible to apply the algorithm to tissues in which individual cells may have only two neighbours, although such examples have not been included in our analysis.

In microscopy videos including division events we expect the algorithm to perform well in tissues in which no adjacent divisions occur between successive movie frames, and in which cells adjacent to the dividing cell do not undergo rearrangements before the next frame is captured. Our algorithm is designed to identify mother and daughter cells of a division event by establishing which are the bordering cells that gain an edge during the division event. In the case of two adjacent divisions, and if cells adjacent to a division event gain edges due to cell rearrangements, the dividing cell cannot be correctly identified. An example of a typical tracking error for two adjacent divisions is shown in figure 10. In cases where the division resolution step fails, our Python implementation returns all tracked cells of the post-processing step, and gives a warning that the division has not been resolved. In these cases, manual correction methods could be used for incorrectly tracked cells in the vicinity of division events.

The parameters of the algorithm are chosen to maximise the robustness of the algorithm and avoid the necessity to adjust the parameters to individual applications. For example, the cutoff length, $d_{\max}$, that determines the distance below which two cells in consecutive movie frames are considered mappable to each other was chosen at 10 times the average cell length in the tissue, which is significantly larger than the movement that is to be expected between consecutive frames of a live-imaging microscopy video. However, parameter adjustments may

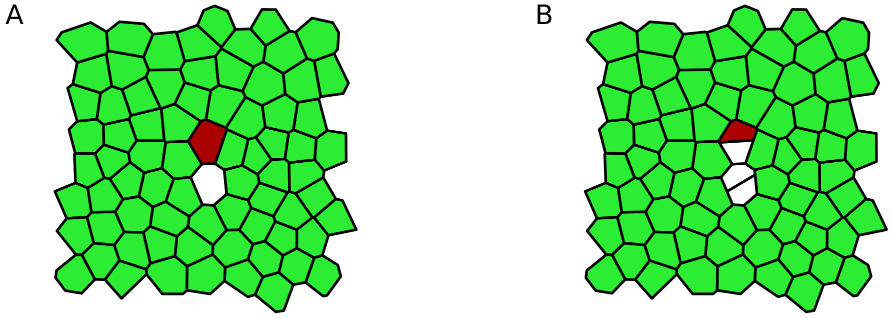

Figure 10: Tracking errors can occur if adjacent cells divide. Here, all green (light) cells are tracked correctly. One of the mother cells (red/dark) of the division events has been incorrectly associated with one of the daughter cells of the division.

be possible for individual applications in order to decrease the algorithm calculation times. For example, the size of the extended neighbourhood considered in the initial step or the iterative extension could be reduced to include only nearest neighbours instead of nearest neighbours and second nearest neighbours in case the tissue is sufficiently heterogeneous. Similarly, one might decrease the cutoff length, $d_{\max}$, for possible cell pairings if the cell positions are not expected to vary significantly between time frames. When filling in any gaps in the MCS (figure 2C-D), requiring that $n_p = 4$ neighbours are preserved between consecutive frames for newly added cells seems to be a sufficiently large number in practice. However, it is possible to start the post-processing requiring a higher number of previously mapped neighbours for the tracking.

Adjustments may be possible to extend the applicability of the algorithm to a wider range of tissues. For example, instead of automatic detection of the initial seeds for the MCS detection algorithm, a small set of seeds could be manually supplied to guide the tracking. This should improve the performance of the algorithm on homogeneous tissues. A further option to improve tracking of cells in homogeneous tissues is to make adaptations that ensure that the cyclic order of neighbours of each cell is preserved under the mapping. In such cases, irregular boundaries may also help to aid the initial seeding. Finally, in cases where the algorithm is not able to track all cells due to a larger number of cell neighbour exchanges than in the tissues considered here, it may be possible to extend the current algorithm by removing the assumption that the MCS is connected, and using a set of non-adjacent cells as initial seeds.

In the presented work, we have deliberately kept the geometrical input to the algorithm to a minimum. Cases where geometric data are taken into account comprise division events where one of the daughter cells is four- or three- sided, since in these cases we are not able to make a decision on which cell is the second daughter cell based on network adjacency alone. If future applications reveal cases where the algorithm performs poorly due to a large number of cell neighbour exchanges or high degree of tissue homogeneity, it may be possible to construct algorithms that combine information on the network topology with data on cell shapes, cell positions and cell movements to improve performance. For example, information on network topology could be integrated into previous algorithms which minimise differences between geometric properties of cells, such as cell size and location [24], with information about network connectivity.

In cell tracking applications, the scaling of the algorithm with tissue size is crucial. Potential applications range from systems of 30 cells (*Drosophila* embryonic epidermal P compartments [9]), to 10,000 cells (*Drosophila* imaginal wing disc [31]). Calculation times in the presented algorithm scale subquadratically with cell number, making it suitable for applications of varying sizes. For example, extrapolating the data in figure 8, a tissue of 10,000 cells could be tracked across two frames within 20 minutes. The scaling of the algorithm is polynomial despite the fact that it is based on MCS detection, which is known to be an NP-hard problem in the general case, i.e. the calculation times scale exponentially with the problem size. MCS detection has a wide range of research applications, including protein interaction networks [48, 49] and finding the binding sites of chemical structures [38]. Our approach of reducing the MCS search to a local, close to quadratically scaling search may have applications in other areas where the networks are inherently planar.

Our algorithm is designed to track cells in segmented microscopy videos of epithelial sheets in two dimensions. However, it may be possible to apply the algorithm to datasets of epithelial sheets that are embedded in a three-dimensional environment, such as the *Drosophila* imaginal wing disc [4], or the *Drosophila* embryonic epidermis [6,9], including tissues that can be mapped onto a cylinder or ellipsoid, such as the mouse visceral endoderm [50].

A large number of cell tracking algorithms have been developed for varying applications [10–27]. Further efforts are required to compare these algorithms with our own, and to identify the algorithm best suited for an individual data set. In the cell tracking challenge [51] the authors provide microscopy videos from a variety of *in vitro* cell cultures, including, for example, mouse embryonic stem cells and human squamous lung carcinoma cells, together with ground-truth segmentation and tracking data as benchmarks for cell tracking and segmentation algorithms. However, many of the published algorithms above have not yet been applied to the challenge, and benchmark datasets for epithelial sheets are currently not available. In [52] *in silico* data sets are used as benchmarking data sets for particle tracking algorithms.

The proposed algorithm provides a tracking solution specialised for cell-tracking in epithelial sheets that attempts to maximise the information that can be gained from the packing that is typical to epithelial tissues. It may, however, be possible to extend this algorithm to applications of two-dimensional cell tracking where cells are not physically connected by constructing adjacency networks from Voronoi tessellations that use the cell locations as seeds. We hope that, as segmentation tools are developed further, the combination of our algorithm with these tools will lead to further insights into cellular behaviour in epithelial tissues.

# Acknowledgements

J.K. acknowledges funding from the Engineering and Physical Sciences Research Council through a studentship. J.Z. acknowledges funding support from the National Science Foundation (awards CBET-1553826 and CBET- 1403887). The authors thank Cody Narciso for sharing the microscopy data.

# References

[1] Stephens, D. J. and Allan, V. J. Light microscopy techniques for live cell imaging. Science, 300(5616):82–86, 2003. 10.1126/science.1082160.

[2] Pantazis, P. and Supatto, W. Advances in whole-embryo imaging: a quantitative transition is underway. Nat. Rev. Mol. Cell. Biol., 15(5):327–339, 2014. 10.1038/nrm3786.

[3] Truong, T. V. and Supatto, W. Toward high-content/high-throughput imaging and analysis of embryonic morphogenesis. Genesis, 49(7):555–569, 2011. 10.1002/dvg.20760.

[4] Mao, Y., Tournier, A. L., Bates, P. A., Gale, J. E., Tapon, N., and Thompson, B. J. Planar polarization of the atypical myosin dachs orients cell divisions in *Drosophila*. Genes Dev., 25(2):131–136, 2011. 10.1101/gad.610511.

[5] Gibson, M. C., Patel, A. B., Nagpal, R., and Perrimon, N. The emergence of geometric order in proliferating metazoan epithelia. Nature, 442(7106):1038–1041, 2006. 10.1038/nature05014.

[6] Rauzi, M., Verant, P., Lecuit, T., and Lenne, P.-F. Nature and anisotropy of cortical forces orienting *Drosophila* tissue morphogenesis. Nat. Cell Biol., 10(12):1401–1410, 2008. 10.1038/ncb1798.

[7] Collinet, C., Rauzi, M., Lenne, P.-F., and Lecuit, T. Local and tissue-scale forces drive oriented junction growth during tissue extension. Nat. Cell Biol., 17(10):1247–1258, 2015. 10.1038/ncb3226.

[8] Ritsma, L., Ellenbroek, S. I. J., Zomer, A., Snippert, H. J., de Sauvage, F. J., Simons, B. D., Clevers, H., and van Rheenen, J. Intestinal crypt homeostasis revealed at single-stem-cell level by *in vivo* live imaging. Nature, 507(7492):362–365, 2014. 10.1038/nature12972.

[9] Parker, J. Control of compartment size by an EGF ligand from neighboring cells. Curr. Biol., 16(20):2058–2065, 2006. 10.1016/j.cub.2006.08.092.

[10] Mashburn, D. N., Lynch, H. E., Ma, X., and Hutson, M. S. Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues. Cytometry A, 81A(5):409–418, 2012. 10.1002/cyto.a.22034.

[11] Cilla, R., Mechery, V., Hernandez de Madrid, B., Del Signore, S., Dotu, I., and Hatini, V. Segmentation and tracking of adherens junctions in 3D for the analysis of epithelial tissue morphogenesis. PLoS Comput. Biol., 11(4):e1004124, 2015. 10.1371/journal.pcbi.1004124.

[12] Schiegg, M., Hanslovsky, P., Kausler, B., Hufnagel, L., and Hamprecht, F. Conservation tracking. IEEE Int. Comp. Vis., 2928–2935, 2013. 10.1109/ICCV.2013.364.

[13] Heller, D., Hoppe, A., Restrepo, S., Gatti, L., Tournier, A., Tapon, N., Basler, K., and Mao, Y. EpiTools: An open-source image analysis toolkit for quantifying epithelial growth dynamics. Dev. Cell, 36(1):103–116, 2016. 10.1016/j.devcel.2015.12.012.

[14] Padfield, D., Rittscher, J., and Roysam, B. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. Med. Image Anal., 15(4):650–668, 2011. 10.1016/j.media.2010.07.006.

[15] Youssef, S., Gude, S., and Radler, J. O. Automated tracking in live-cell time-lapse movies. Integr. Biol., 3:1095–1101, 2011. 10.1039/C1IB00035G.

[16] Wait, E., Winter, M., Bjornsson, C., Kokovay, E., Wang, Y., Goderie, S., Temple, S., and Cohen, A. Visualization and correction of automated segmentation, tracking and lineaging from 5-D stem cell image sequences. BMC Bioinform., 15(1):328, 2014. 10.1186/1471-2105-15-328.

[17] Winter, M., Wait, E., Roysam, B., Goderie, S. K., Ali, R. A. N., Kokovay, E., Temple, S., and Cohen, A. R. Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing. Nat. Protocols, 6(12):1942–1952, 2011. 10.1038/nprot.2011.422.

[18] Sommer, C., Straehle, C., Köthe, U., and Hamprecht, F. A. Ilastik: Interactive learning and segmentation toolkit. In 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 230–233. 2011. 10.1109/ISBI.2011.5872394.

[19] Liu, K., Lienkamp, S. S., Shindo, A., Wallingford, J. B., Walz, G., and Ronneberger, O. Optical flow guided cell segmentation and tracking in developing tissue. In 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI), 298–301. 2014. 10.1109/ISBI.2014.6867868.

[20] Bellaiche, Y., Bosveld, F., Graner, F., Mikula, K., Remesikova, M., and Smisek, M. New robust algorithm for tracking cells in videos of *Drosophila* morphogenesis based on finding an ideal path in segmented spatio-temporal cellular structures. In IEEE Annu. Int. Conf. Eng. Med. Biol. Soc., 6609–6612. 2011. 10.1109/IEMBS.2011.6091630.

[21] Aly, A. A., Deris, S. B., and Zaki, N. Intelligent algorithms for cell tracking and image segmentation. Int. J. Comput. Sci. Inf. Technol., 6(5):21–37, 2014.

[22] Wang, Q., Niemi, J., Tan, C.-M., You, L., and West, M. Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy. Cytometry A, 77A(1):101–110, 2010. 10.1002/cyto.a.20812.

[23] Raffel, M., Willert, C. E., Wereley, S., and Kompenhans, J. Particle Image Velocimetry: A Practical Guide. Springer-Verlag Berlin Heidelberg, 2007.

[24] Puliafito, A., Hufnagel, L., Neveu, P., Streichan, S., Sigal, A., Fygenson, D. K., and Shraiman, B. I. Collective and single cell behavior in epithelial contact inhibition. Proc. Natl. Acad. Sci. U.S.A., 109(3):739–744, 2012. 10.1073/pnas.1007809109.

[25] Al-Kofahi, Y., Lassoued, W., Lee, W., and Roysam, B. Improved automatic detection and segmentation of cell nuclei in histopathology images. IEEE Trans. Biomed. Eng., 57(4):841–852, 2010. 10.1109/TBME.2009.2035102.

[26] Amat, F., Lemon, W., Mossing, D. P., McDole, K., Wan, Y., Branson, K., Myers, E. W., and Keller, P. J. Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data. Nat. Meth., 11(9):951–958, 2014. 10.1038/nmeth.3036.

[27] Aigouy, B., Farhadifar, R., Staple, D. B., Sagner, A., Röper, J.-C., Jülicher, F., and Eaton, S. Cell flow reorients the axis of planar polarity in the wing epithelium of *Drosophila*. Cell, 142(5):773–786, 2010. 10.1016/j.cell.2010.07.042.

[28] Hoebe, R. A., Van Oven, C. H., Gadella, T. W. J., Dhonukshe, P. B., Van Noorden, C. J. F., and Manders, E. M. M. Controlled light-exposure microscopy reduces photobleaching and phototoxicity in fluorescence live-cell imaging. Nat. Biotech., 25(2):249–253, 2007. 10.1038/nbt1278.

[29] Wood, W. and Jacinto, A. Cell Migration: Developmental Methods and Protocols, chapter Imaging Cell Movement During Dorsal Closure in *Drosophila* Embryos, 203–210. Humana Press, Totowa, NJ, 2005. 10.1385/1-59259-860-9:203.

[30] Mavrakis, M., Rikhy, R., Lilly, M., and Lippincott-Schwartz, J. Fluorescence Imaging Techniques for Studying *Drosophila* Embryo Development. John Wiley & Sons, Inc., 2001. 10.1002/0471143030.cb0418s39.

[31] Farhadifar, R., Röper, J.-C., Aigouy, B., Eaton, S., and Jülicher, F. The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing. Curr. Biol., 17(24):2095–2104, 2007. 10.1016/j.cub.2007.11.049.

[32] Escudero, L. M., da F. Costa, L., Kicheva, A., Briscoe, J., Freeman, M., and Babu, M. M. Epithelial organisation revealed by a network of cellular contacts. Nat. Commun., 2:526, 2011. 10.1038/ncomms1536.

[33] Sánchez-Gutiérrez, D., Tozluoglu, M., Barry, J. D., Pascual, A., Mao, Y., and Escudero, L. M. Fundamental physical cellular constraints drive self-organization of tissues. EMBO J., 2015. 10.15252/embj.201592374.

[34] Sáez, A., Acha, B., Montero-Sánchez, A., Rivas, E., Escudero, L. M., and Serrano, C. Neuromuscular disease classification system. J. Biomed. Opt., 18(6):066017–066017, 2013. 10.1117/1.JBO.18.6.066017.

[35] Ullmann, J. R. An algorithm for subgraph isomorphism. JACM, 23(1):31–42, 1976. 10.1145/321921.321925.

[36] Krissinel, E. B. and Henrick, K. Common subgraph isomorphism detection by backtracking search. Software Pract. Exper., 34(6):591–607, 2004. 10.1002/spe.588.

[37] Wilson, R. An Introduction to Graph Theory. Prentice Hall, 5th edition, 2010.

[38] Raymond, J. W. and Willett, P. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. J. Comput.-Aided Mol. Des., 16(7):521–533, 2002. 10.1023/A:1021271615909.

[39] Osborne, J. M., Bernabeu, M. O., Bruna, M., Calderhead, B., Cooper, J., Dalchau, N., Dunn, S.-J., Fletcher, A. G., Freeman, R., Groen, D., et al. Ten simple rules for effective computational research. PLoS Comput. Biol., 10(3):e1003506, 2014. 10.1371/journal.pcbi.1003506.

[40] Hagberg, A. A., Schult, D. A., and Swart, P. J. Exploring network structure, dynamics, and function using NetworkX. In Proceedings of the 7th Python in Science Conference (SciPy2008), 11–15. Pasadena, CA USA, 2008.

[41] Honda, H. Description of cellular patterns by Dirichlet domains: The two-dimensional case. J. Theor. Biol., 72(3):523–543, 1978. 10.1016/0022-5193(78)90315-6.

[42] Narciso, C., Wu, Q., Brodskiy, P., Garston, G., Baker, R., Fletcher, A., and Zartman, J. Patterning of wound-induced intercellular Ca 2+ flashes in a developing epithelium. Phys. Biol., 12(5):056005, 2015. 10.1088/1478-3975/12/5/056005.

[43] Parton, R. M., Vallés, A. M., Dobbie, I. M., and Davis, I. Collection and mounting of *Drosophila* embryos for imaging. Cold Spring Harbor Protocols, 2010(4):pdb.prot5403, 2010. 10.1101/pdb.prot5403.

[44] Nagai, T., Kawasaki, K., and Nakamura, K. Vertex dynamics of two-dimensional cellular patterns. J. Phys. Soc. Jpn., 57(7):2221–2224, 1988. 10.1143/JPSJ.57.2221.

[45] Etournay, R., Popović, M., Merkel, M., Nandi, A., Blasse, C., Aigouy, B., Brandl, H., Myers, G., Salbreux, G., Jülicher, F., et al. Interplay of cell dynamics and epithelial tension during morphogenesis of the *Drosophila* pupal wing. eLife, 4:e07090, 2015. 10.7554/eLife.07090.

[46] Warn, R. and Magrath, R. F-actin distribution during the cellularization of the drosophila embryo visualized with FL-phalloidin. Exp. Cell Res., 143(1):103 – 114, 1983. http://dx.doi.org/10.1016/0014-4827(83)90113-1.

[47] Classen, A.-K., Anderson, K. I., Marois, E., and Eaton, S. Hexagonal packing of *Drosophila* wing epithelial cells by the planar cell polarity pathway. Dev. Cell, 9(6):805–817, 2005. 10.1016/j.devcel.2005.10.016.

[48] Ciriello, G., Mina, M., Guzzi, P. H., Cannataro, M., and Guerra, C. Alignnemo: A local network alignment method to integrate homology and topology. PLoS ONE, 7(6):e38107, 2012. 10.1371/journal.pone.0038107.

[49] Aladağ, A. E. and Erten, C. Spinal: scalable protein interaction network alignment. Bioinformatics, 29(7):917–924, 2013. 10.1093/bioinformatics/btt071.

[50] Trichas, G., Smith, A. M., White, N., Wilkins, V., Watanabe, T., Moore, A., Joyce, B., Sugnaseelan, J., Rodriguez, T. A., Kay, D., et al. Multi-cellular rosettes in the mouse visceral endoderm facilitate the ordered migration of anterior visceral endoderm cells. PLoS Biol., 10(2):e1001256, 2012. 10.1371/journal.pbio.1001256.

[51] Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., España, T., Venkatesan, S., Balak, D. M., et al. A benchmark for comparison of cell tracking algorithms. Bioinformatics, 30(11):1609–1617, 2014. 10.1093/bioinformatics/btu080.

[52] Chenouard, N., Smal, I., de Chaumont, F., Maska, M., Sbalzarini, I. F., Gong, Y., Cardinale, J., Carthel, C., Coraluppi, S., Winter, M., et al. Objective comparison of particle tracking methods. Nat. Meth., 11(3):281–289, 2014. 10.1038/nmeth.2808.