Automated Cell Lineage Construction: A Rapid Method to Analyze Clonal Development Established with Murine Neural Progenitor Cells

Omar Al-Kofahi, Richard J. Radke, Susan K. Goderie, Qin Shen, Sally Temple and Badrinath Roysam

Preprint of article to appear in *Cell Cycle*, Vol. 5, No. 3, Feb. 1 2006, pp. 327-335

Abstract

Understanding cell lineage relationships is fundamental to understanding development, and can shed light on disease etiology and progression. We present a method for automated construction of lineages of proliferative, migrating cells from a sequence of images. The method is applicable to image sequences gathered either in vitro or in vivo. Currently, generating lineage trees from progenitor cells over time is a tedious, manual process, which limits the number of cell measurements that can be practically analyzed. In contrast, the automated method is rapid and easily applied, and produces a wealth of measurements including the precise position, shape, cell-cell contacts, motility and ancestry of each cell in every frame, and accurate timings of critical events, e.g., mitosis and cell death. Furthermore, it automatically produces graphical output that is immediately accessible. Application to clonal development of mouse neural progenitor cells growing in cell culture reveals complex changes in cell cycle rates during neuron and glia production. The method enables a level of quantitative analysis of cell behavior over time that was previously infeasible.

INTRODUCTION

Cell tracking over time has been one of the most revealing types of study for understanding developmental mechanisms. Normal progenitor cells undergo complex processes, including cell division, migration, changes in morphology, and death that are critical for tissue formation. Moreover, pathologically changed cells, e.g., cancer cells, exhibit specific changes in these processes that correlate with disease progression. In order to follow cell behavior over time, researchers have found ways to record them in time-lapse images. However, analysis of these images is an arduous process involving many hours of human inspection to extract even simple measurements. Most of the information contained in the sequence cannot be extracted because it is impossible for an observer to accurately follow many different events over the image sequence. Our aim in this study was to design an automated method to track and quantify cell behavior in image sequences that would be widely applicable to cells either in vivo or in vitro.

Classical studies of development of a simple animal, *C. elegans*, described the divisions and movements of identified progenitor cells, revealed by direct observation through the transparent body wall. They showed that individual progenitors undergo distinct division patterns correlated with changes in gene expression, morphology, migration, spatial location and ultimately cell differentiation or death (Sulston 1976, Sulston 1977, Chalfie 1981, Sulston 1981). Moreover, mutations were identified that altered progenitor cell behavior directly observed by microscopy, providing fundamental information about genes critical for many aspects of normal development (Chalfie 1981, Sulston 1981). These basic methods of watching cells have been applied to a variety of developing systems, and form the framework for understanding how tissues form and how they are changed in disease.

While vertebrate embryos lack advantages of *C. elegans* embryos such as simplicity and transparency, which allow visual tracking of progenitors in vivo, their progenitor cells can be followed continually ex vivo. For example, the behavior of germinal neural cells can be time-lapse recorded from slices (Haydar 1999, De Marchis 2001, Noctor et

al. 2001, Nadarajah 2001, Miyata 2001) or from dissociated cultures (Rivas et al. 1995). We showed previously that mouse neural progenitors growing as clones have surprisingly similar lineage trees to those of *C. elegans*, implying evolutionary conservation of mechanisms for generating neural cells (Qian 1998). Moreover, neural stem cells, which can generate both neuronal and glial progeny, undergo a characteristic pattern of cell divisions over time, in which early divisions are overtly asymmetric and produce neurons while later divisions are overtly symmetric and produce glial progeny (Qian 2000). Thus isolated neural stem cells growing in tissue culture are able to recapitulate the normal order of production of cells in vivo, where most neurons are made before glia. We suggest that the divisions of vertebrate neural progenitor cells over time are, like their invertebrate counterparts, highly regulated, and play a critical role in the production of diverse neural cell types.

These studies used manual lineage tree reconstruction. This is a slow and painstaking process in which individual cells are tracked by an observer and followed as they migrate, divide, differentiate, die or move out of the field. The arduousness of the process precludes analyzing a large number of cells, and hampers investigation of complex aspects of clonal development, such as changes in cell cycle, cell-cell contacts, or motility. While these events can be observed in the images, it is not possible to efficiently record and quantify them manually. Consequently, we cannot easily assess the effects of environment or genetic factors on many aspects of clonal development, as these experiments require high-throughput analysis to compare many different cells. To enable these studies, we have now developed an automated lineage construction method to analyze recorded image sequences. We believe that this paper represents the first description of a fully automatic system for accurately lineaging progenitor cells through many divisions and thousands of frames. We note that (Braun et al. 2003) described a tool for analyzing and visualizing existing lineages, but did not address automatic lineage construction from images. (Yasuda et al. 1999) described a semi-automatic algorithm for lineaging *C. elegans* cells from the 2-cell to 7-cell stage, but this system required a substantial amount of human intervention to deal with the large number of falsely detected and undetected cells.

There are two general approaches to automated tracking of cells in time-series images. The first approach involves segmenting cells in each image frame independently based on properties such as intensity, texture and gradient (Wu 1995, Wahlby 2004), followed by an association of cells between consecutive image frames (Kirubarajan 2000). This approach is simple and efficient, but cell segmentation errors (e.g., multiple touching cells that are mistakenly segmented as one, or cells that are undetected) can seriously affect tracking results. In addition, performing the association in a manner that accounts for cell division can be problematic (Kirubarajan 2000).

The second approach involves segmenting the cells in the first frame, and then tracking these cells (e.g., updating their locations and orientations) throughout the rest of the sequence. Approaches based on active contours (Blake 2000) are often used for this purpose (Dormann 2002, Ray 2002, Zimmer 2002). In cases where obtaining the exact cell boundary is not critical, the mean-shift algorithm (Collins 2003) gives a fast solution for tracking cells. The key difficulty with this class of methods is that they do not naturally handle cell division, and additional heuristics

are required to initialize a new contour when a split occurs (Debeir 2005). Level set methods (Sethian 1999, Dufour 2005) are able to handle topology changes due to cell division; however, deciding which cell is the parent of each child would still be an issue with such methods. Generally speaking, tracking many single cells independently would produce poor results in many situations, both in vivo and in vitro, in which there are many cells in a small area that split quickly and sometimes ambiguously.

We designed our cell-tracking algorithm based on the first approach. Every image in the time series is preprocessed to remove non-uniform illumination effects, and then thresholded adaptively (Otsu 1979) to segment the cells. We used a seeded watershed algorithm (Vincent 1991) to overcome segmentation errors caused by touching cells. We also designed a multiple-object matching method that can handle cell divisions, dead cells, and cell segmentation errors. Given a time-lapse image sequence, regions corresponding to cells are automatically segmented in each image of the sequence, and then matched over time as cells deform, move, divide, and die. Along the way, our algorithm measures a number of cell attributes such as size, shape and aspects of morphology, location, motility and migration, and relations between multiple cells, such as cell-cell contacts. We developed this approach using recordings of mouse neural progenitor cell divisions. However, it is widely applicable to image sequences obtained of different cell types in many different contexts, in vivo or ex vivo, e.g., in slices or in dissociated cultures. In fact, the technique could be applied to any image sequence of objects that migrate and/or divide, including whole organisms or sub-cellular organelles.

MATERIALS AND METHODS

Isolation of single cortical progenitor cells. Cerebral cortices were dissected from embryos of timed-pregnant Swiss Webster mouse (Taconic Farms) at embryonic day E10-E12 (plug date is designated day 0) and papain dissociated, as described previously (Qian 1998). Briefly, the tissue was rocked for 30 minutes in DMEM with L-glutamine, sodium pyruvate and N-acetyl-cysteine (NAC, Sigma) plus papain and DNAse (Sigma) at room temperature, then spun at 300 g for 10 min and rinsed with DMEM. The tissue was triturated, the cell suspension was settled for 15 minutes and the top fraction which contains $> 95\%$ single cells was collected for plating.

Cell culture. The single cell suspension was plated into poly-L-lysine coated Terasaki plate microwells in serum-free culture medium: DMEM with L-gutamine, sodium pyruvate, B-27, N-2 (all from Gibco), NAC and supplemented with 10 ng/ml bFGF (Gibco). Cultures were maintained in a humidified tissue culture incubator at $35°$C with 6% CO_2.

Time-lapse video microscopy and manual rescontruction of lineage trees. Plated cells were placed under an Olympus inverted microscope in a humidified chamber at $35°$C with 6% CO_2. Phase images were captured by a CCD camera (Panasonic BL600) and recorded with a time-lapse video cassette recorder (Panasonic). Cultures were monitored for 37 days, then fixed and stained for cell fate markers: β-tubulin III for neurons, O4 for oligodendrocytes and GFAP for astrocytes. For manual lineage tree reconstruction, the video taped recordings were replayed and the

movements and divisions of individual cells were followed. Progeny of a single cell were tracked to generate the lineage tree and their final fates were identified using the corresponding immunostaining data.

Immunostaining of embryonic cortical neural cells. After recording clonal development, the clones were fixed and stained to identify cell progeny. For identifying oligodendrocyte- lineage cells, live cultures were incubated with O4 antibody (ATCC hybridoma) for 20 minutes at $35°$C, then washed with calcium-magnesium-containing PBS (CMPBS) and fixed with 4% paraformaldehyde (PFA) for 30 minutes. Fixed cultures were blocked with 10% normal goat serum in PBS for 15 minutes at room temperature, then incubated with goat-anti-mouse IgM at 1:100 in PBS for 45 minutes at room temperature. For identifying neurons, fixed cells were permeablized with methanol at $-20°$C for 5 min before adding β-tubulin III antibody (1:100, Sigma) overnight at $4°$C. After rinsing and blocking, fluroescein (FITC)- or rhodamine-conjugated goat-anti-mouse IgG was added to reveal β-tubulin III staining.

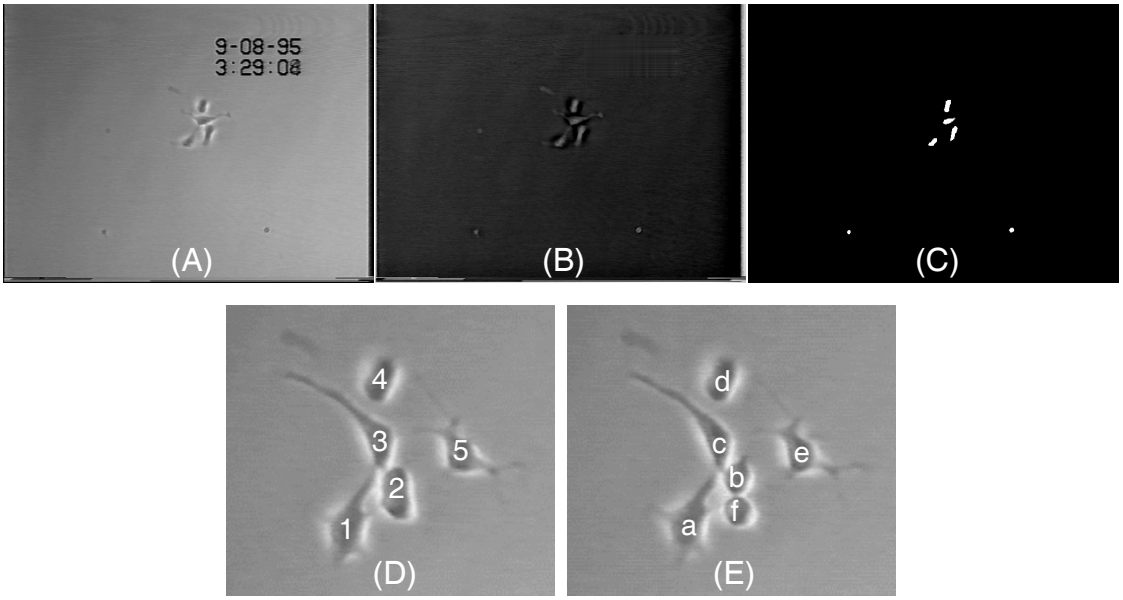
Automated reconstruction of lineage trees. All the image analysis was implemented with Matlab 7.0 (MathWorks Inc.), and was run on a PC computer with an AMD Opteron 2.2 GHz processor, 4 GB of memory, and the Gentoo Linux operating system (Gentoo Foundation, Inc.).

RESULTS

Development of automated lineage construction for neural progenitor cells. Nineteen individual murine embryonic cerebral cortical progenitor cells that had been followed over several days using time-lapse microscopy formed the basis of this study. The clone development, described as a basic lineage tree, had been previously manually reconstructed from time-lapse video recordings by two independent observers with agreement, and provided basic ground truth for comparison with subsequent automated tracking. In one typical field, designated Sequence A, four cells were tracked over three days (Supplementary Movie 1). The images were then processed using our automated lineage construction algorithm.

The videotaped images were digitized with a frame grabber (Fig. 1A) to yield a sequence of images. In the description that follows, we use "image t" to indicate one frame from this sequence. The subsequent frame is referenced as "image $(t+1)$". The video recordings contained a time-stamp that is potentially confusing to automated analysis systems. This was outlined manually in the first frame and removed automatically in the rest of the sequence. A homomorphic filter (Phong 1975) was used to correct the nonuniform illumination present in the images (Fig. 1B). The progenitor cells were labeled manually in the first frame with one mouse-click per cell (this too can be automated in principle). This labeling is the only manual intervention required in the lineaging process, aside from a semi-automatic training phase described below that can be applied to a large number of image sequences from similar studies.

Each image is thresholded with an intensity value estimated adaptively (Otsu 1979) to produce a binary image of segmented cells as illustrated in Figure 1C. Segmented objects with size less than ten pixels were considered as noise and rejected. Each segmented cell was approximated with an ellipse for subsequent matching computations.



	Event	Likelihood
	1→a	**0.86**
	1→b	0.18
	1→c	0.15
	1→f	0.16
	2→b	0.36
	2→f	0.35
	2→c	0.22
L	**3→c**	**0.84**
	3→f	0.20
	4→d	**0.87**
	5→e	**0.88**
	1→a,b	0.07
	2→b,f	**0.84**
	3→c,f	0.12

(F)

M					*N*					
0.43	**0**	**0**	**0**	**0**	**0.43**	**0**	**0**	**0**	**0**	**0**
0.09	0	0	0	0	0	0.09	0	0	0	0
0.08	0	0	0	0	0	0	0.08	0	0	0
0.08	0	0	0	0	0	0	0	0	0	0.08
0	0.18	0	0	0	0	0.18	0	0	0	0
0	0.18	0	0	0	0	0	0	0	0	0.18
0	0.11	0	0	0	0	0	0.11	0	0	0
0	**0**	**0.42**	**0**	**0**	**0**	**0**	**0.42**	**0**	**0**	**0**
0	0	0.10	0	0	0	0	0	0	0	0.10
0	**0**	**0**	**0.46**	**0**	**0**	**0**	**0**	**0.46**	**0**	**0**
0	**0**	**0**	**0**	**0.44**	**0**	**0**	**0**	**0**	**0.44**	**0**
0.02	0	0	0	0	0.02	0.02	0	0	0	0
0	**0.28**	**0**	**0**	**0**	**0**	**0.28**	**0**	**0**	**0**	**0.28**
0	0	0.04	0	0	0	0	0.04	0	0	0.04

(G)

Fig. 1. *Cell segmentation and hypothesis generation.* (A) One video image frame of the neural progenitor cell culture from Sequence *A*, shown fully in Supplementary Movie 1. (B) The result of pre-processing to remove the time stamp by linearly interpolating intensities from local pixels. The illumination variation was also removed using homomorphic filtering, and the result was inverted for display. (C) The result of adaptive thresholding for the processed image in panel B. (D, E) A pair of consecutive images in which one cell (cell 2) undergoes mitosis. (F) Several matches between cells in (D) and (E) detected as feasible, with their likelihoods. Correct matches are highlighted in bold, showing that their likelihoods are high. (G) Several rows of the matrix A used for integer programming.

Cells that are either close to each other, or are actually touching, are often segmented as one blob using this method. Such large blobs were split into component cells using the seeded watershed algorithm (Vincent 1991). We further constrain the centers of the parts to occur at the centers of overlapping cells in the previous image, which mitigates most problems with oversegmentation. We note that at this stage of the algorithm, other more sophisticated cell segmentation algorithms could be substituted if warranted. However, we demonstrate below that this straightforward segmentation scheme produced high-accuracy results for the murine cells of interest in this study.

For each segmented cell, we calculate several simple features that suffice to characterize it for the purposes of associating cells between images. Specifically, for cell i in image t, we compute its centroid (x_i^t, y_i^t), area a_i^t, eccentricity e_i^t, major axis length l_i^t, and orientation θ_i^t. These six numbers are collected into a vector $f_i^t = (x_i^t, y_i^t, a_i^t, e_i^t, l_i^t, \theta_i^t)$ that we call the feature vector for cell i at time t. We consider three possible events per cell that can take place between time t and time $t+1$. The cell can either: (1) deform and move; (2) divide into two new cells; or (3) die. To decide among these hypotheses, we need to match cells in image t with cells in image $t+1$.

We model the likelihood that the cell labeled as i in frame t (with feature vector f_i^t) deforms and moves to become the cell labeled as j in frame $t+1$ (with feature vector f_j^{t+1}) as:

$$p_{move}(f_i^t, f_j^{t+1}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} \exp\left\{-\frac{1}{2}(d_{ij}^t - \mu)^T \Sigma^{-1} (d_{ij}^t - \mu)\right\}, \quad (1)$$

where $N = 6$ and d_{ij}^t is the absolute difference between the feature vectors, $d^t ij = |f_i^t - f_j^{t+1}|$ (i.e., the vector of absolute differences between corresponding cell characteristics). The parameters μ and Σ are prior estimates of the mean and covariance of the difference vector d_{ij}^t for correctly matching cells. These parameters can be semi-automatically estimated by computing the sample mean and covariance of a correctly labeled training set of corresponding cells. For example, in the experiments described in this paper, we used a training sequence of 270 images, with an average of six cells per image. This sequence was chosen based on a few minutes of visual inspection to make sure cells exhibited characteristic motions. These cells were automatically segmented and tracked with a simple nearest-neighbor algorithm (with the user making a few edits to the tracking results when matching errors occurred). The feature vectors of each cell were computed and used to generate the sample mean and covariance (μ, Σ). The same (μ, Σ) values were subsequently used to process the thousands of frames in multiple image sequences described in this paper. As long as the imaging setup and the cell behaviors are similar to the training sequence, there is no need to adjust these parameters.

The likelihood of the hypothesis that f_i^t divided into two cells f_j^{t+1} and f_k^{t+1} is estimated in a similar manner: f_j^{t+1} and f_k^{t+1} are first merged into f_{jk}^{t+1} by fitting an ellipse to the union of the two corresponding segmented cells, and the absolute difference between f_i^t and f_{jk}^{t+1} is used in a likelihood computation as in equation (1):

$$p_{divide}(f_i^t, f_{jk}^{t+1}) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} \exp\left\{-\frac{1}{2}(d_{ijk}^t - \mu)^T \Sigma^{-1} (d_{ijk}^t - \mu)\right\}, \quad (2)$$

where $N = 6$ and $d^t_{ijk} = |f^t_i - f^{t+1}_{jk}|$. Hence, we can generate the likelihood for any possible hypothesis, be it (cell to cell) or (one cell to two cells); some examples are shown in Fig. 1DG. From this figure, one can see that the correct behaviors (e.g., $1 \rightarrow a$ and $2 \rightarrow (b, f)$) generally have high likelihoods, while incorrect behaviors (e.g., $1 \rightarrow b$) do not. Matches we know to be correct based on manual ground truth are shown in bold. Our goal is to determine the optimal (i.e., highest total likelihood) set of correspondences between cells in image t and image $t + 1$, subject to the constraints that (1) each cell in image t has at most two daughter cells in image $t + 1$, and (2) no cell in image $t + 1$ has more than one parent cell in image t. We do not enforce the constraint that every cell in both images has to be accounted for, since the segmentation results may contain false detections that do not represent actual cells, or miss valid cells. These segmentation errors are handled after solving the matching problem, as described below.

We also addressed the cell death hypothesis in the problem formulation. When a cell dies, it usually stops moving and begins to disintegrate, leaving an observable ghost. Eventually the ghost can disappear entirely, but this usually takes many days in clonal density cultures where macrophage activity is negligible. In the image sequences processed to date, dead cells slowly disintegrated into smaller objects, each of which is eventually below the size threshold for segmentation. Thus, at some point, the cell is segmented in one image and not in the next. We discuss in more detail below how to distinguish cell death from an image segmentation error.

We formulated the matching problem as follows. Given M cells at image t, and N cells at image $t + 1$, we construct a matrix A of size $L \times (M + N)$, where each of the L rows corresponds to a feasible move or divide hypothesis. A subset of this matrix is illustrated in Fig. 1G; the first M columns correspond to the cells at time t, and the remaining N columns correspond to cells at time $t + 1$. For a row that represents a correspondence between cell feature vectors f^t_i and f^{t+1}_j (e.g., the first row in Fig. 1G), the likelihood in (1) is evenly split between columns i and $M + j$ of this row, and the remaining columns are set to zero. For a row that represents the division of cell f^t_i into f^{t+1}_j and f^{t+1}_k (e.g., the last row in Fig. 1G), the likelihood in (2) is evenly split between columns i, $M + j$, and $M + k$ of this row, while the remaining columns are set to zero.

Finding the optimal set of matches between cells in image t and cells in image $t + 1$ corresponds to finding a subset of rows in A such that the sum of the entries in these rows is maximized, under the constraint that no two rows share common nonzero entries. Mathematically, this is posed as the following integer programming problem:

$$\max_x \ (Av)^T x \qquad s.t. \quad B^T x \leq v, \tag{3}$$

where x is a binary $L \times 1$ vector (1 if the row is in the solution, 0 if not), v is a vector of ones of size $(M + N) \times 1$, and B is a binary matrix of the same size as A, with $B(i, j) = 1$ if $A(i, j) > 0$ and $B(i, j) = 0$ otherwise. The solution x is found using integer programming (Wolsey 1998). Specifically, we used a linear programming (LP)-based branch-and-bound algorithm that searches for an optimal solution by solving a series of LP-relaxation problems, in which the binary integer requirement on the variables is replaced by the weaker constraint $0 < x < 1$.

Finally, we post-process the result to account for various segmentation errors:

1) We allow the number of cells detected in image $t+1$ to increase compared to image t due to cell division only; we do not allow new cells to spontaneously appear. Hence, cells detected in image $t+1$ that are not matched to cells in image t are considered false positives, and rejected. This implies that cells that enter the field of view of the microscope will be rejected as well, which is not a problem since we are only interested in tracking descendants of cells that were present in the first image in the sequence.
2) Similarly, should a cell leave the field of view of the microscope, it is removed from the lineage analysis, since there is no reliable way to reason about multiple cells that move off the field, possibly divide, and then move back in. In the more than five thousand images processed in our experiments, only 3% of the tracked cells moved off the field of view.
3) If a cell in image t is not matched with any cell in image $t+1$, there are two possibilities. Either the cell is actually present but was undetected in image $t+1$ due to a segmentation error, or the cell is dead and there actually is no match in image $t+1$. We distinguish between these two possibilities by copying the position and shape of the cell in image t to image $t+1$, but labeling this cell as "missed at time t". If the missed cell is matched by the integer program in subsequent frames, we know the earlier miss was due to a momentary segmentation error. However, if the missed cell continues to be propagated forward with no match for many frames (in our example, we used 40 minutes from the miss at time t as a threshold), we mark the cell as dead, note the time of death as time t, and stop searching for it in subsequent frames.
4) In addition to missed cells and false positives, a single cell in image $t+1$ can be detected as two due to segmentation errors. We must ensure that we can reliably distinguish cell division from oversegmentation. We observed that when a murine neural progenitor cell divides, the two daughter cells have similar sizes. Thus, whenever cell division is detected, a significance test is applied to the difference between the areas of the two daughter cells. That is, we assume that the division is correct if

$$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(a_j^{t+1} - a_k^{t+1})^2\right\} > \tau, \tag{4}$$

where a_j^{t+1} and a_k^{t+1} are the areas of the two daughter cells. Otherwise, we detect an oversegmentation and merge the two pieces into a single cell. The variance σ^2 is estimated from divisions in the training sequence; in our experiments, we used a significance level of 0.2, corresponding to $\tau = 0.8$.

The computation time required to solve the matching problem is proportional to the number of competing hypotheses. To reduce computation time, we do not consider hypotheses that are highly unlikely (e.g., $1 \rightarrow d$ in Fig. 1). A likelihood threshold of 0.1 was used for the results in this paper. Consequently, the number of hypotheses is usually only slightly larger than the number of cells in image $t+1$.

Figure 2 illustrates the result of automatic lineaging for Sequence A with 2100 frames. Four progenitor cells are present in the first frame, which have divided into 43 cells by the last frame 71 hours later. The total computation time was 37 minutes for cell segmentation and 46 minutes for cell matching, i.e., less than 3 seconds per frame. The

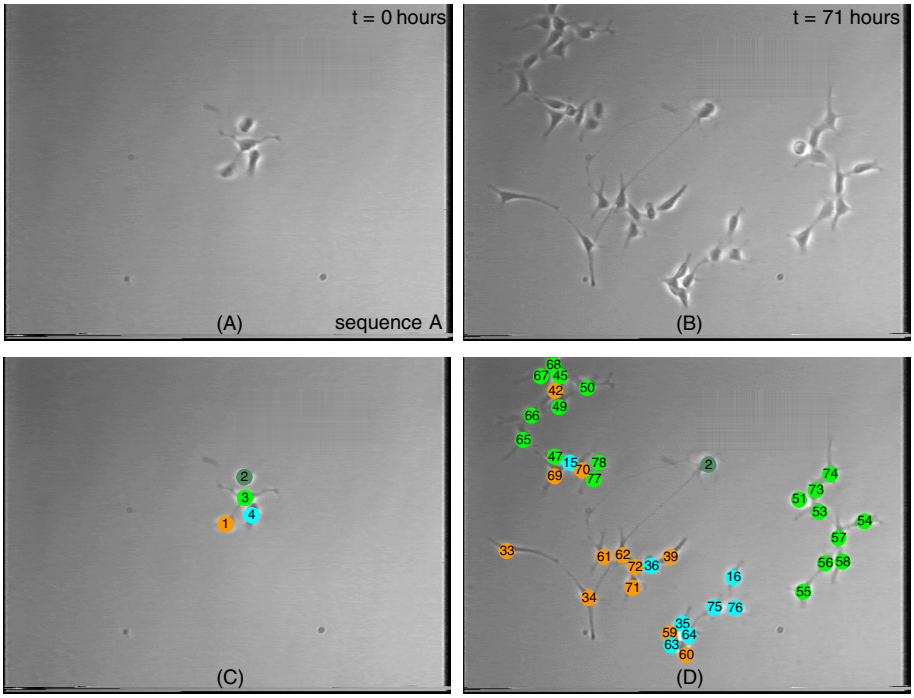


Fig. 2. *Raw and automatically annotated frames from Sequence A.* (A-B). Video frames of Sequence A at times 0 (A) and 71 hours (B) are shown. (C-D). The corresponding automatically annotated frames of (A, B). At the first frame (C), each cell was assigned a number and color. The daughter cells inherit the display color of their ancestor and are numbered automatically according to their birth order. In the final frame (D), cells with the same color are derived from a single cell. Supplementary Movie 2 illustrates the entire annotated result.

top row of images shows the first and last frames of the sequence, while the bottom row shows the automatically segmented and lineaged cells. In panel (D), the color of a cell corresponds to that of its ancestor in the first frame. Cells are also numbered to indicate the order in which they appeared. Figure 3 shows automatically constructed lineage diagrams for each progenitor in Sequence A. The red number at each branch point indicates the time of division in hours. The blue number on each vertical line indicates the cell cycle time in hours (and the length of the line is proportional to this number). Dashed lines indicate cells that continued to divide, but were not analyzed. The lineages were verified to be correct by comparing with manual ground truth (which was not accessed while developing the computer algorithms). Supplementary Movie 2 shows the lineage relationships shown as colored, numbered dots overlaid on the video image sequence. The process was repeated for a second sequence of time-lapse images of 5 progenitor cells (Fig. 4, Supplementary Movies 3 and 4). In this case, three cell deaths were identified (Fig. 4E). Again, the error rate was negligible, and the resulting trees are similar to those already published (Qian 1998, Qian 2000). For example, neuroblasts (progenitors restricted to making neurons) had either short, overtly symmetric lineages (Fig. 4G-I) or longer asymmetric lineages (Fig. 3C, cell 4; Supplementary Fig. 2F). In contrast, glioblasts (progenitors restricted to making glia) were more proliferative and generated overtly symmetric lineage

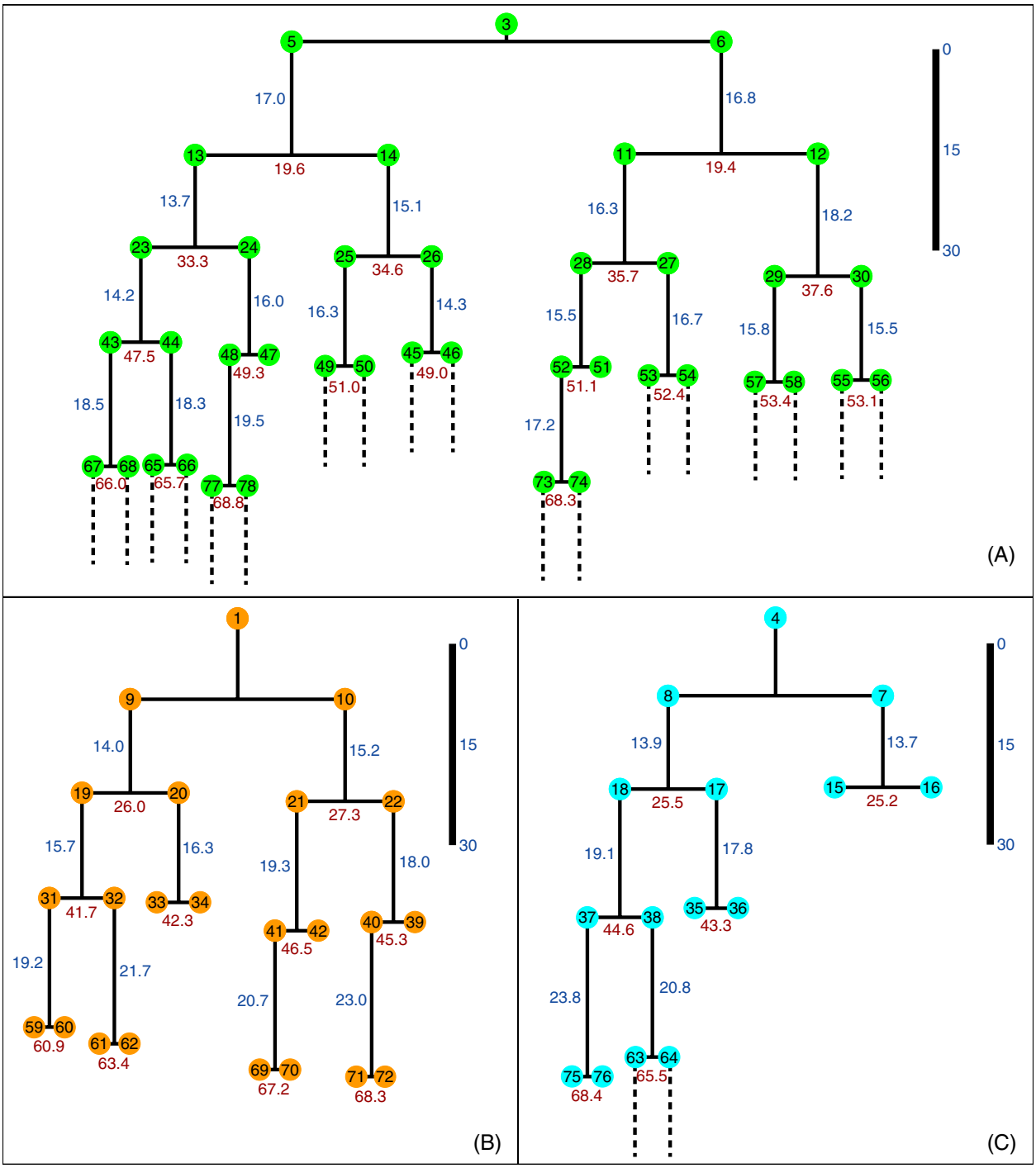


Fig. 3. *Automatically generated lineage trees for Sequence A*. The red number at each branch point indicates the time of division in hours. The blue number on each vertical line indicates the cell cycle time in hours (the length of the line is proportional to this number; see scale bar in hours). Dashed lines indicate cells that continued to divide after the automated tracking was stopped. The color of each numbered cell corresponds to the color in Figure 2, and Supplementary Movie 2. (A) Cell 3 gave rise to a proliferative, overtly symmetrical lineage tree reflecting a typical glial cell lineage. (B, C) Cell 1 and Cell 4 generated typical asymmetric neuronal lineage trees.

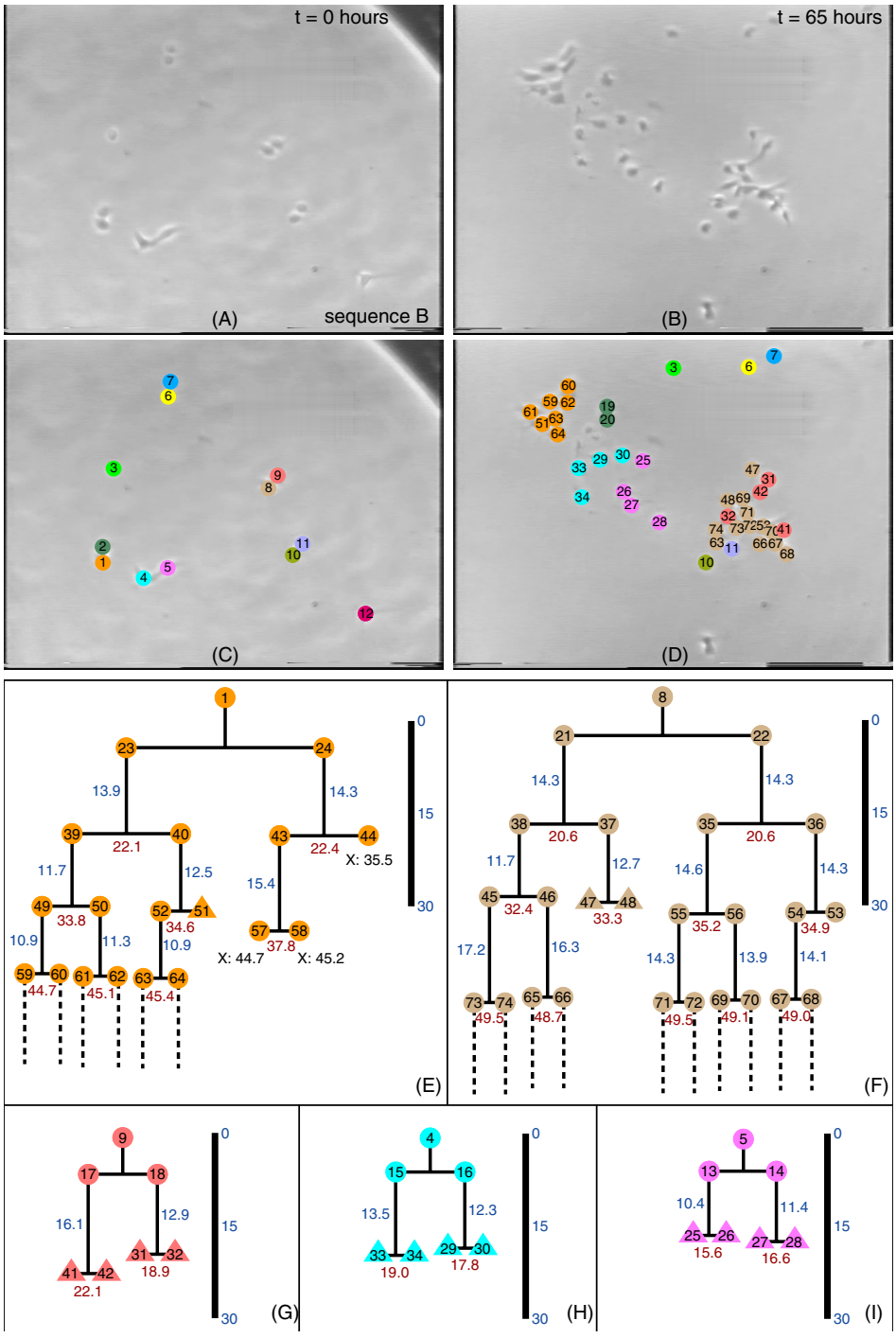


Fig. 4. *Automated analysis of Sequence B*. (A-B). Video frames of Sequence B at times 0 (A) and 65 hours (B). (C-D) The corresponding automatically annotated frames of (A) and (B). (E-I). The automatically generated lineage results of cells that divided (Cell 1, 8, 9, 4, 5). Triangles in the corresponding lineage tree indicate β-tubulin III positive neurons. Cells labeled with an "X" were automatically detected to have died at the indicated time (the number next to X in hours). Supplementary Movie 4 illustrates the entire annotated result.

trees (Fig. 3A). Stem cells or multipotent progenitors first undergo asymmetric divisions to produce neurons and then more proliferative divisions to produce glial progeny (Fig. 4F).

Neural progenitor cells exhibit different cell cycle rates. Our automated lineage construction algorithm allows detailed measurement of cell cycle rates as the clone develops. While we had recorded changes in division mode during clonal development (Qian 1998, Qian 2000), we had not previously investigated whether this was accompanied by changes in cell cycle dynamics.

Population studies of the neural germinal zones in vivo have shown that the cell cycle of the neural progenitor population lengthens considerably during embryonic development. Most neocortical neurons are formed between embryonic days 11-17 in the mouse. Birth of cells is accompanied by an increase in cell cycle to approximately 8-18 hours (Caviness 2003). In addition, neuroblasts have a longer cell cycle rate than nonneuronal progenitor cells (Calegari 2005). However, it has not yet been possible to study changes in cell cycle over time within an individual clone. Nor is it known if the varieties of neuronal or glial restricted progenitors or stem cells have different cell cycle rates. By gathering the information from a number of different lineage trees, we found that cell cycle was more heterogeneous than previously recognized, that different types of progenitors had characteristic cell cycle dynamics, and that there are changes in cell cycle rates correlated with neuron and glia production.

Figure 5 illustrates automatically generated statistics of cell cycle times computed from the embryonic murine cortical progenitor cell lineages in Figure 3 for Sequence A. Figures 5B-D illustrate the time of divisions starting from each progenitor cell, while Figure 5A illustrates the combined division times for all three active progenitors. Figure 6 shows the same results presented in a graph where the horizontal axis is real time, allowing us to see when specific cells, e.g. siblings, divided. We disregarded the time period for each initial plated cell to divide, as this did not comprise a complete cell cycle. In Supplementary Figures 1 and 2, we illustrate the same type of automated analysis for two additional image sequences (Supplementary Movies 3, 4 and 5, 6).

For these early cortical progenitor cells, we found that in many cases, cell cycle lengthens as the division number increases. This is especially obvious for neuroblasts (Fig. 5B and D; Supplementary Fig. 2F). Hence this process, detected at the population level in vivo, is recapitulated in vitro within individual clones. Most (for example 78% of sequences A and B) of the cell cycles we recorded were within the 8–18 hours seen in vivo. However, there were some clones (e.g., Supplementary Fig. 2) that had significantly longer cell cycle times; this particular neuroblast had one progeny that divided after almost four days. This is consistent with the observation that cells specified to be neuronal progenitors have longer cell cycle times (Calegari 2005, Takahashi 1995), although such long cycles have not been described previously. It is possible that some cells with significantly longer division rates exist in germinal zones in vivo, because the behavior of small sub-populations of cells can be missed in population labeling studies. Moreover, we know that progenitor cells in adult germinal zones adopt very long cell cycle rates (Kippin 2005).

Glioblasts showed less obvious slowing of cell cycle during the recording period. In fact, we observed in some a

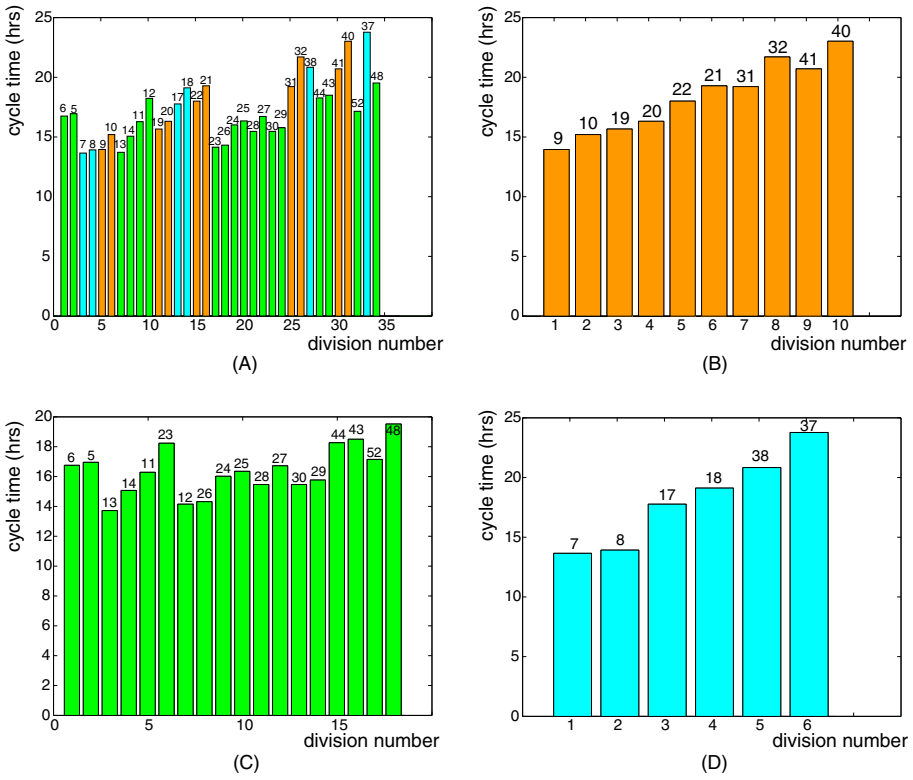


Fig. 5. *Automatically measured cell cycle times for Sequence A as a function of division number*. Divisions are numbered in ascending order of culture time. The number atop each bar indicates the cell number in Figure 2, and in Supplementary Movie 2. (A) All cells aggregated. Different colors correspond to different clones. (B-D) Each cell was separately plotted. (B) cell 1, (C) cell 3. (D) cell 4. Note that in (B) and (D), which are typical neuroblast lineages, the cell cycle lengthens over division numbers. In contrast, (C), a typical glioblast, shows a varied pattern with less obvious cell cycle lengthening and some points where cell cycle speeds up, e.g. after the first division.

speeding up of the cell cycle, especially after the first one or two divisions in vitro. Whether this reflects a response to exogenous growth factors applied in vitro is not clear. However, the contrast in behavior with neuroblasts is marked, and suggests substantially different cell cycle regulatory mechanisms in these two classes of progenitor cells.

The cell cycle is more complex for multipotent progenitor cells. As observed previously, neurons arose from asymmetric divisions occuring early in the lineage tree, and then the cell switched over to glial cell production. We noted that the cell cycle can transiently increase around the time when the stem cell switches from making neurons to making glia, similar to the speeding up observed in glioblasts (Fig. 4E, cell 1).

Despite the heterogeneity of cell cycle lengths exhibited by different neural progenitors, we found that siblings often divided within a short time of each other, as illustrated in Figure 6, which reveals rather close synchronicity in mitoses of lineally related cells. This is supported by Figure 7, which shows via linear regression that siblings

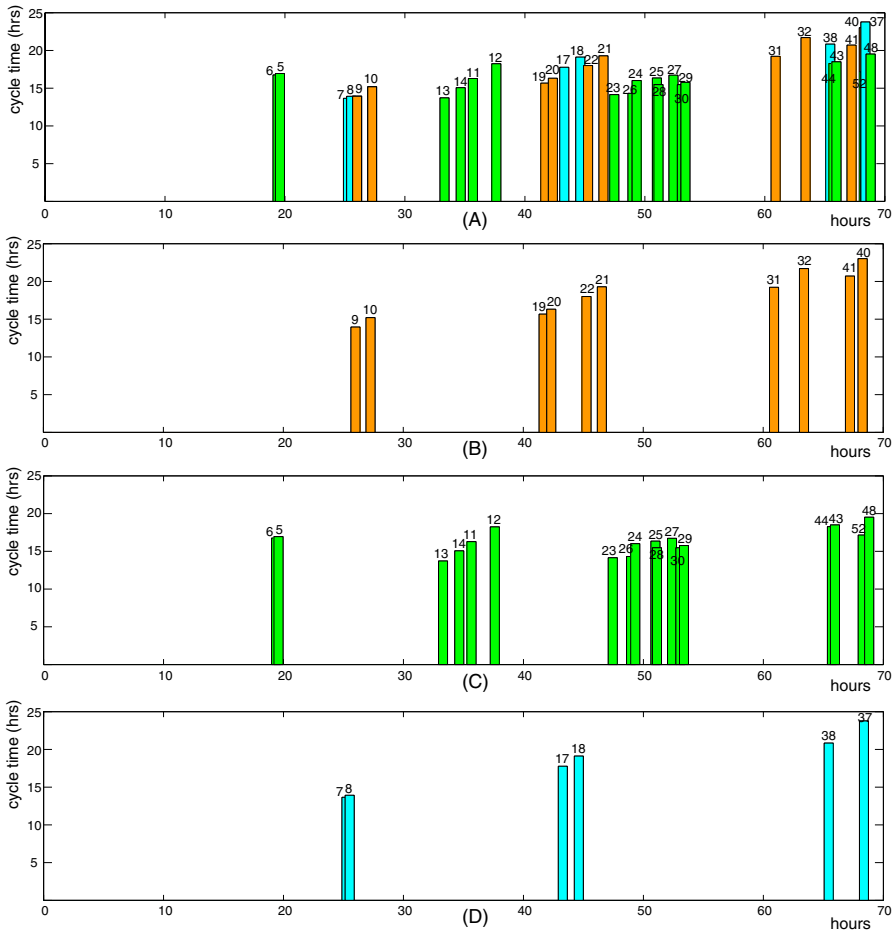


Fig. 6. *Automatically measured cell cycle times for Sequence A as a function of real time.* (A) All cells aggregated. (B) cell 1, (C) cell 3, (D) cell 4. Cells are identified by numbers atop bars. Siblings have consecutive numbers. We note a high degree of synchronicity between siblings, with related cells clearly dividing at similar time-points across the sequence.

have similar cell cycle lengths (the linear regression $R^2 = 0.91$). These data indicate that lineage relationship plays a role in determining division rate. Clearly, cell cycle rate is a highly dynamic and regulated aspect of cell behavior during clonal development, likely reflecting different states of progenitor cell commitment.

DISCUSSION

This study shows that it is possible to accurately automate cell lineage construction, and to derive from this novel information regarding proliferative cell behavior over time. Since the computer vision algorithms are very fast (i.e., only 23 seconds per frame in our experiments), the automatic lineaging process can be considered to be real time with respect to the image acquisition speed- that is, the lineaging results can be made immediately available while the cells are dividing. Multiple lineages can be constructed simultaneously, allowing on-time analysis of a field or

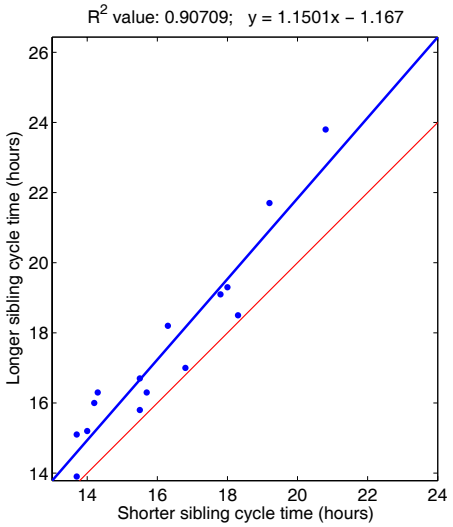


Fig. 7. *Linear regression indicates sibling cell cycle times are similar*. Sibling cell cycle length was plotted with the shorter cell cycle time of the pair as the X-axis and the longer cell cycle time of the pair as the Y-axis. Linear regression on the cell cycle times for siblings in Sequence A reveals a definite trend, with an R^2 value of 0.91. This indicates sibling cells have highly related cell cycle times, indicating a heritable aspect to cell cycle control.

array of cells. While in this study we focused on changes in cell cycle rate, other distinct aspects of the clone behavior can be extracted, e.g., cell motility or cell-cell contacts.

As discussed above, a small amount of training data representative of the sequences to be analyzed is required to set the parameters of the algorithm (e.g., (μ, Σ) in (1) and (2), (σ, τ) in (4)), and some manual intervention may be required in this phase. This intervention, which amounts to a few simple mouse clicks to correct cell matching errors, is not burdensome, especially compared to the tedious task of manually lineaging numerous image sequences. In the future, we plan to design intuitive user interfaces to further simplify the training process.

Improved technology for microscopy and digital image acquisition will make the automatic lineaging even more robust. For example, the murine image sequences were digitized from archived, 9-year old videotape. This was a good starting point because the lineage trees had already been manually constructed for these cells, but image quality was not optimal. We are currently acquiring high-resolution digital images, which are highly preferable to the videotaped images in that they (1) have much better contrast (i.e., the cells stand out more strongly from the background), (2) have fewer image acquisition artifacts (e.g., time stamps, VCR captions, image noise), (3) are higher-resolution and inherently digital, and (4) were acquired with the intention of processing them with computer vision algorithms. We further expect our general approach to be applicable to more advanced imaging modalities (e.g., fluorescence, confocal, or two-photon microscopy), as well as 4D image stacks.

We note that different types of cells may require modifications of the move and divide hypotheses discussed

above. For example, *Drosophila* or *C. elegans* progenitor cells can divide into cells of different sizes, which would require us to adjust the assumption we made about murine cells. The automatic lineaging algorithm can also benefit from additional mathematical models describing cell shapes or division patterns when they are available. We also acknowledge that there are some cell behaviors that will always be nearly impossible to cope with, with either a manual or automatic approach. For example, in some cases, multiple cells merge into a single, undifferentiated blob for many frames and eventually split apart- there is no natural way to track individual cells once they enter the blob, so accurate lineaging would be very difficult. In such cases, there is little recourse but to terminate the lineage analysis. We have found that some cell types and culture conditions exacerbate blob formation, and expect that improved culture conditions will mitigate this problem.

In conclusion, automation of lineage construction allows high-throughput analysis of clonal development, enabling studies of how environmental and genetic factors impact cell development over time. It also enables in-depth studies of dynamic cell populations, providing the means to gather and quantitate complex cell behaviors and cell-cell interactions observable over a sequence of images. Most importantly, the automated method makes possible studies that were previously infeasible due to the number of cells that need to be processed, the precision of morphometric or timing measurements that are required, or the multi-frame spatial relationships that need to be analyzed. Finally, the general applicablility of this method allows quantification of any image sequence of objects that split and/or migrate. We are currently applying the framework to time-lapse sequences of *C. elegans* progenitor cell lineages in vivo, and to *C. elegans* whole organisms, to analyze and quantify animal behavior. As techniques to image live cells in vivo in vertebrates are developed, this automated method can be applied to analyze the behavior of endogenous or transplanted cells in tissues.

ACKNOWLEDGEMENTS

We thank Brian Lewis for help with digital image acquisition and instrumentation. This work was supported by grant R37NS033529 from NINDS and by CenSSIS, the NSF Center for Subsurface Sensing and Imaging Systems, under the award EEC-9986821.

REFERENCES

1) Blake A, Isard M. Active contours: The application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion. London: Springer, 2000.
2) Braun V, Azevedo RB, Gumbel M, Agapow PM, Leroi AM, Meinzer HP. ALES: Cell lineage analysis and mapping of developmental events. Bioinformatics 2003; 19:851-8.
3) Calegari F, Haubensak W, Haffner C, Huttner WB. Selective lengthening of the cell cycle in the neurogenic subpopulation of neural progenitor cells during mouse brain development. J Neurosci 2005; 25:6533-8.

4) Caviness Jr VS, Goto T, Tarui T, Takahashi T, Bhide PG, Nowakowski RS. Cell output, cell cycle duration and neuronal specification: A model of integrated mechanisms of the neocortical proliferative process. Cereb Cortex 2003; 13:592-8.
5) Chalfie M, Horvitz HR, Sulston JE. Mutations that lead to reiterations in the cell lineages of C. elegans. Cell 1981; 24:59-69.
6) Collins R. Mean-shift blob tracking through scale space. Computer Vision and Pattern Recognition. 2003.
7) Debeir O, Van Ham P, Kiss R, Decaestecker C. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. IEEE Trans Med Imaging 2005; 24:697-711.
8) De Marchis S, Fasolo A, Shipley M, Puche A. Unique neuronal tracers show migration and differentiation of SVZ progenitors in organotypic slices. J Neurobiol 2001; 49:326-38.
9) Dormann D, Libotte T, Weijer CJ, Bretschneider T. Simultaneous quantification of cell motility and protein-membrane-association using active contours. Cell Motil Cytoskeleton 2002; 52:221-30.
10) Dufour A, Shinin V, Tajbakhsh S, Guillen-Aghion N, Olivo-Marin JC, Zimmer C. Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces. IEEE Trans Image Process 2005; 14:1396-410.
11) Haydar TF, Bambrick LL, Krueger BK, Rakic P. Organotypic slice cultures for analysis of proliferation, cell death, and migration in the embryonic neocortex. Brain Res Brain Res Protoc 1999; 4:425-37.
12) Kippin TE, Martens DJ, van der Kooy D. p21 loss compromises the relative quiescence of forebrain stem cell proliferation leading to exhaustion of their proliferation capacity. Genes Dev 2005; 19:756-67.
13) Kirubarajan T, Bar-Shalom Y, Pattipati KR. Multiassignment for tracking a large number of overlapping objects. Norwood, MA: Artech House, 2000.
14) Miyata T, Kawaguchi A, Okano H, Ogawa M. Asymmetric inheritance of radial glial fibers by cortical neurons. Neuron 2001; 31:727-41.
15) Nadarajah B, Brunstrom JE, Grutzendler J, Wong RO, Pearlman AL. Two modes of radial migration in early development of the cerebral cortex. Nat Neurosci 2001; 4:143-50.
16) Noctor SC, Flint AC, Weissman TA, Dammerman RS, Kriegstein AR. Neurons derived from radial glial cells establish radial units in neocortex. Nature 2001; 409:714-20.
17) Otsu N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics 1979; 9:62-66.
18) Phong B. Illumination for computer generated pictures. Communications of the ACM 1975; 18:311-17.
19) Qian X, Goderie SK, Shen Q, Stern JH, Temple S. Intrinsic programs of patterned cell lineages in isolated vertebrate CNS ventricular zone cells. Development 1998; 125:3143-52.
20) Qian X, Shen Q, Goderie SK, He W, Capela A, Davis AA, Temple S. Timing of CNS cell generation: A programmed sequence of neuron and glial cell production from isolated murine cortical stem cells. Neuron

2000; 28:69-80.

21) Ray N, Acton ST, Ley K. Tracking leukocytes in vivo with shape and size constrained active contours. IEEE Transactions on Medical Imaging 2002; 21:1222-35.

22) Rivas RJ, Hatten ME. Motility and cytoskeletal organization of migrating cerebellar granule neurons. J Neurosci 1995; 15:981-9.

23) Sethian JA. Level set methods and fast marching methods. Cambridge, UK: Cambridge Univ. Press, 1999.

24) Sulston JE. Post-embryonic development in the ventral cord of Caenorhabditis elegans. Philos Trans R Soc Lond B Biol Sci 1976; 275:287-97.

25) Sulston JE, Horvitz HR. Post-embryonic cell lineages of the nematode, Caenorhabditis elegans. Dev Biol 1977; 56:110-56.

26) Sulston JE, Horvitz HR. Abnormal cell lineages in mutants of the nematode Caenorhabditis elegans. Dev Biol 1981; 82:41-55.

27) Takahashi T, Nowakowski RS. Caviness Jr VS. Early ontogeny of the secondary proliferative population of the embryonic murine cerebral wall. J Neurosci 1995; 15:6058-68.

28) Vincent L, Soille P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans Pattern Analysis Machine Intelligence 1991; 13:583-98.

29) Wahlby C, Sintorn IM, Erlandsson F, Borgefors G, Bengtsson E. Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections. J Microsc 2004; 215:67-76.

30) Wolsey LA. Integer programming. New York: J Wiley, 1998.

31) Wu K, Gauthier D, Levine MD. Live cell image segmentation. IEEE Trans Biomed Eng 1995; 42:1-12.

32) Yasuda T, Bannai H, Onami S, Miyano S, Kitano H. Towards automatic construction of cell-lineage of C. elegans from nomarski DIC microscope images. Genome Inform Ser Workshop Genome Inform 1999; 10:144-54.

33) Zimmer C, Labruyere E, Meas-Yedid V, Guillen N, Olivo-Marin JC. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing. IEEE Trans Med Imaging 2002; 21:1212-21.

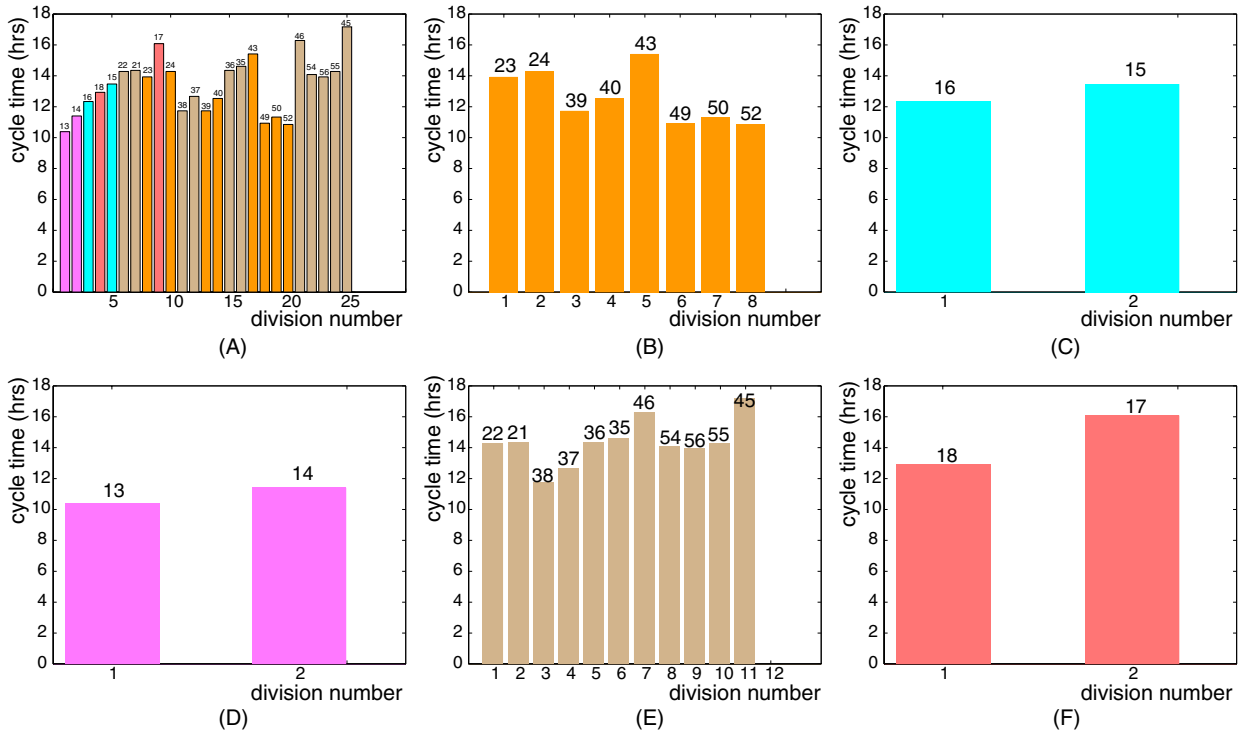


Fig. 8. *Automatically measured cell cycle times for Sequence B*. Divisions are ordered in ascending order of division time. (A) All cells aggregated. (B) cell 1, (C) cell 4, (D) cell 5, (E) cell 8, (F) cell 9. The number atop each bar indicates the cell number in Figure 4, and in Supplementary Movie 4.

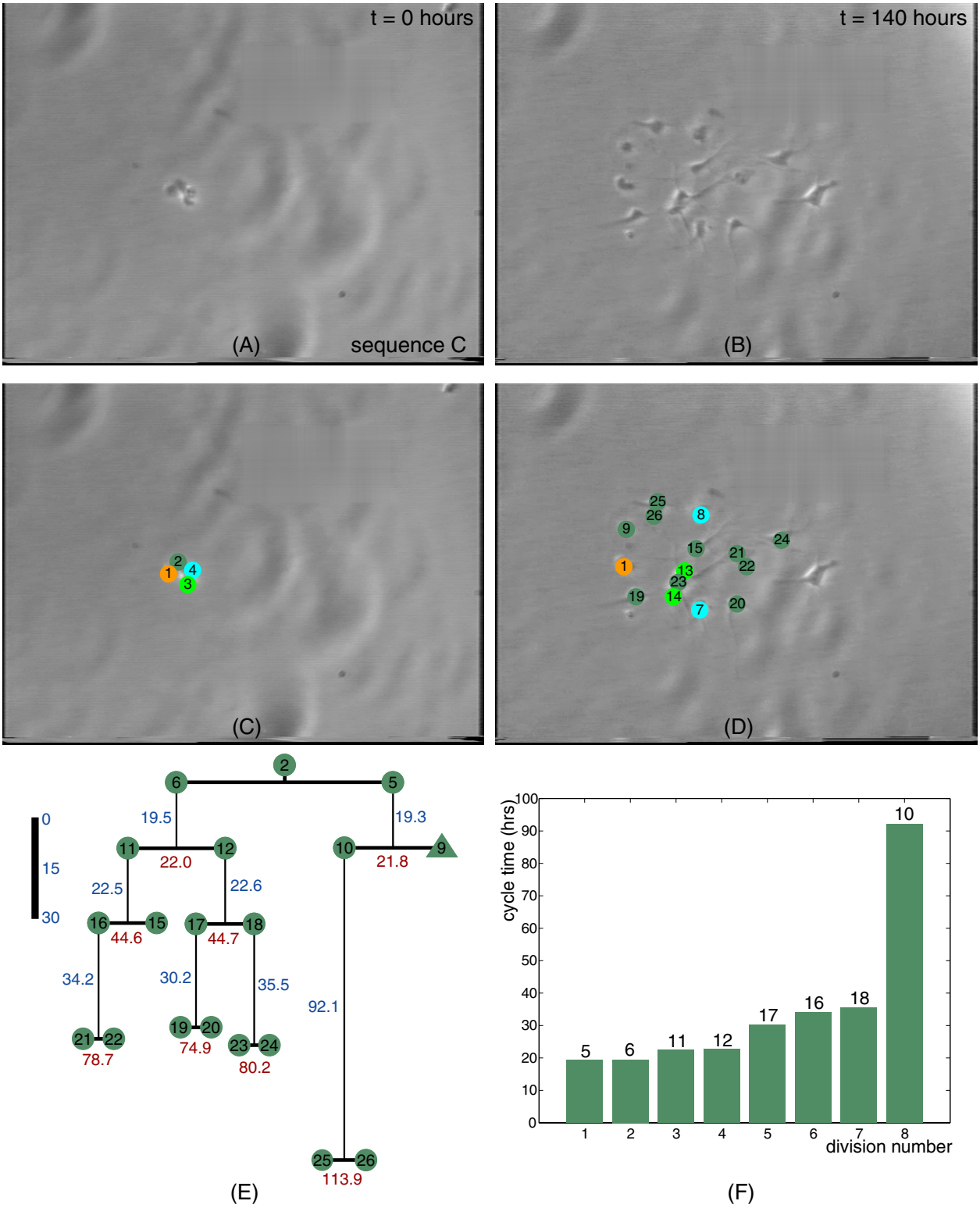


Fig. 9. *Automated analysis of Sequence C*. Frames of Sequence C at times 0 and 140 hours are shown in the upper panels (A-B). The corresponding automatically generated lineage results for cell 2 are shown on the lower panels (C-F). The triangle in the lineage tree indicates a neuron. Note the unusually long cycle time for cell 10. Supplementary Movie 6 illustrates the entire annotated result.

SUPPLEMENTARY MOVIES

All movies can be downloaded from the *Cell Cycle* website at `http://www.landesbioscience.com/journals/cc/supplement/alkofahi.zip`.

The original sequence is provided, followed by a second movie in which the cells are automatically overlaid with colored dots that indicate the ancestry of the cell. The number of each dot indicates the order of division for that sequence. Please note that while detailed morphometry is extracted by the algorithm for each cell, it is not illustrated here for clarity.

Supplementary Movie 1

Original captured images, Sequence A. This sequence has a total of 2100 frames. Four progenitor cells are present in the first frame, which have divided into 43 cells by the last frame 71 hours later.

Supplementary Movie 2

Automatically lineaged images, Sequence A.

Supplementary Movie 3

Original captured images, Sequence B. This sequence has a total of 700 frames. Twelve progenitor cells are present in the first frame, which have divided into 42 cells by the last frame 65 hours later.

Supplementary Movie 4

Automatically lineaged images, Sequence B.

Supplementary Movie 5

Original captured images, Sequence C. This sequence has a total of 2100 frames. Four progenitor cells are present in the first frame, which have divided into 13 cells by the last frame 140 hours later.

Supplementary Movie 6

Automatically lineaged images, Sequence C.