Ordered, Random, Monotonic, and Non-monotonic Digital Nanodot Gradients

Grant Ongo, [1,2] *Sébastien G. Ricoult,* [2,3] *Timothy E. Kennedy,* [3] *and David Juncker* [1,2,3, *]

[1]Department of Biomedical Engineering, McGill University,
740 Dr. Penfield Avenue, Montreal, Quebec, Canada H3A 0G1
[2]Genome Quebec Innovation Centre, McGill University, Montréal, Quebec, Canada H3A 0G1
[3]McGill Program in Neuroengineering, Department of Neurology and Neurosurgery, Montreal Neurological Institute, McGill University, 3801 University Ave,
Montreal, Quebec, Canada H3A 2B4

Corresponding Author
David Juncker
Dept. Biomedical Engineering
Micro and Nanobioengineering Laboratory
McGill University
740 Dr. Penfield, room 6206
Montreal, Quebec, Canada, H3A 0G1
Phone: (514) 398-7676, Fax: (514) 398 1790
Email: david.juncker@mcgill.ca

ABSTRACT

Cell navigation is directed by inhomogeneous distributions of extracellular cues. It is well known that noise plays a key role in biology and is present in naturally occurring gradients at the micro- and nanoscale, yet it has not been studied with gradients *in vitro*. Here, we introduce novel algorithms to produce ordered and random gradients of discrete nanodots – called digital nanodot gradients (DNGs) – according to monotonic and non-monotonic density functions. The algorithms generate continuous DNGs, with dot spacing changing in two dimensions along the gradient direction according to arbitrary mathematical functions, with densities ranging from 0.02% to 44.44%. The random gradient algorithm compensates for random nanodot overlap, and the randomness and spatial homogeneity of the DNGs were confirmed with Ripley's K function. An array of 100 DNGs, each 400 × 400 μm^2, comprising a total of 57 million 200 × 200 nm^2 dots was designed and patterned into silicon using electron-beam lithography, then patterned as fluorescently labeled IgGs on glass using lift-off nanocontact printing. DNGs will facilitate the study of the effects of noise and randomness at the micro- and nanoscales on cell migration and growth.

Introduction

Gradients are fundamental to many phenomena of biology, from directing axonal navigation during neural development to the differentiation of stem cells in response to an injury [1,2]. Gradients may occur as either (i) free diffusion gradients, (ii) substrate-bound gradients, or (iii) a combination of both [3]. Free diffusion gradients can be generated by cell-excreted proteins that diffuse in the extracellular space. Substrate-bound gradients arise when proteins are bound to the extracellular matrix or cell membranes [4-6]. To better understand cellular haptotaxis – the directed movement of a cell or axonal growth cone along a gradient of a substrate-bound guidance cue – many techniques to generate surface-bound gradients *in vitro* have been developed [7]. Two classes of concentration gradients exist, namely continuous and digital gradients. Continuous gradients have protein concentration changing in a constant manner, and are typically produced by adsorbing molecules from a diffusible gradient. Several parameters of these gradients have been modulated, including the range and slope, which was linear, exponential or non-monotonic [8]. For example, linear gradients of the protein BDNF have been shown to modulate neuron polarization and growth [9] and linear laminin gradients have been shown to orient rat hippocampal axon specification based on slope [10]. Secondly, digital gradients which have been recently introduced, are formed by patterning small dots of protein and varying the density of the dots by changing their size [11], the spacing between them [12], or both [13]. The advantage of digital gradients is that they are deterministic, and that the local concentration can be precisely predicted and does not rely on fluorescence measurements which are prone to error. Digital gradients have been used to study how retinal ganglion cells identify the stop zone within graded distributions of repulsive EphrinA5 ligands [13]. However, digital gradients rarely extend over 1-2 orders of magnitude (OM), whereas it is believed that the dynamic range of gradients *in vivo* is between 3-4 OM. To overcome these limitations, we previously developed digital nanodot gradients (DNGs), where the spacing between nanodots (200 nm in diameter) was changed in two dimensions to produce a dynamic range exceeding 3 OM. These designs were implemented using a low-cost, lift-off nanocontact printing method to pattern substrate-bound gradients of proteins and peptides. We employed these patterns for adhesion and migration studies of C2C12 myoblasts on RGD peptide and netrin-1 gradients, respectively [14]. For these experiments, gradients of 400 × 400 μm^2 were divided into 53 rectangular boxes of fixed size and density. With this approach, gradients were non-continuous and had stepwise changes in density. This is most pronounced in low-density regions where the spacing between nanodots exceeded the dimensions of the box, requiring larger box sizes and thus creating large “steps” at the bottom of the gradient. This may prove problematic as cells may fail to sense a discontinuous gradient if they fall into a region of constant density.

Noise is ubiquitous in biology [15], and modern patterning technologies can be exploited to introduce defined amounts of noise and randomness into otherwise regular patterns. The effect of randomness was evaluated in ordered arrays and repetitive patterns with a constant average density. In one study, disorder was introduced in arrays of 120 nm dots spaced 300 nm apart by randomly displacing the dot by up to 50% of the spacing to avoid overlap [16]. While this approach introduced a controlled amount of noise by restricting the maximum displacement, it was not truly random since dots were each contained within the original grid. Nonetheless, cellular adhesion and stem cell differentiation of osteoblasts were markedly altered as a result of increasing disorder. In another study, whole proteome analysis of cells grown on the same disordered patterns resulted in differential expression of certain proteins in the extracellular signal-regulated kinase (ERK1/2) pathway [17]. Similarly, controlled amounts of topographical noise on nanogratings of 500 nm ridges and grooves has shown to effect PC12 neuronal cell alignment to the gratings, focal adhesion maturation and directionality [18].

Randomness and noise are also highly relevant to directed cell migration. The stochasticity of chemo- and haptotaxis has been well studied, and is apparent from the random-walk like traces of migrating cells [19]. It is well understood that biological gradients, which appear continuous, are in fact quantized since they are comprised of individual molecules adsorbed to a surface. The distribution of these molecules is not deterministic, but stochastic at the nanoscale. The engagement of receptors from migrating cells with these guidance cues has been modeled within a stochastic framework [19,20]. Random variations in the gradient also occur at the microscale *in vivo* from the local accumulation of chemo- and haptotactic molecules that form concentration puncta [21]. Gradients formed by the expression of receptors from a cells embedded in a tissue a may become non-monotonic as particular cells over- and under- express a receptor relative to their position in the gradient [22]. Cells navigating through such a patchwork of microscale deviations must discriminate against local maxima and minima to sense the overarching gradient slope. It has been suggested that cells alternate between periods of sensitization and desensitization and are thereby capable of maintaining an overall response to long-range gradients [23].

Here, we introduce continuous DNGs eliminating the stepwise density changes of the previously reported “step” design [14] with (i) ordered and (ii) random positioning of nanodots, as well as (iii) non-monotonic DNGs that can be implemented using either (i) or (ii). We discuss various strategies to create noisy DNGs and outline the challenges in forming truly random DNGs with accurate slopes. The slope of random DNGs was measured and compared to the programmed density function, and their randomness verified using Ripley’s K-function. We generated an array of 100 ordered and random 400 × 400 μm^2 large DNGs made of 200 × 200 nm^2 nanodots, including monotonic and non-monotonic density curves, with a dynamic ranges spanning from 2.14 to 3.86 OM. Non-monotonic gradients produced here aim to introduce in a quantitative and repeatable manner “microscale noise” into surface-bound *in vitro* gradients. The entire array of 100 DNGs covers a 35 mm^2 area and is comprised of more than 57 million nanodots. The DNG array was patterned onto a silicon (Si) wafer by electron-beam (e-beam) lithography, and was transferred onto glass slides by lift-off nanocontact printing of fluorescently labeled IgGs [14]. The fidelity of the replication process was evaluated by overlaying the DNG design with fluorescence microscopy images of the printed IgG proteins.

Materials and Methods

Software

Algorithms were developed in Matlab R2013a (Natick, MA, USA). Scripts of the ordered and random gradient algorithms are available for download upon request. A spreadsheet template (Microsoft Excel 2010) of gradient parameters was imported into Matlab for processing. The output for each gradient was a text file of coordinates formatted as a Caltech Intermediate Format (CIF) file. CIFs were imported into CleWin Version 4.1 (WieWeb, MESA Research Institute at the University of Twente and Deltamask, Netherlands). Gradients were exported from CleWin as Bitmap (BMP) image files for verification of density using ImageJ 1.47 64-bit (National Institutes of Health, USA) and Matlab. The 100-gradient array was designed in L-edit (Tanner EDA, Monrovia, CA, USA).

Electron-Beam Lithography

A 4” Si wafer was coated with PMMA resist and the 100-gradient array was patterned by e-beam lithography (VB6 UHR EWF, Vistec, Montreal, QC, Canada), followed by reactive-ion etching (System100 ICP380, Plasmalab, Everett, WA, USA) 100 nm deep into the Si wafer.

Stamp Fabrication

After cleaning, the Si wafer was coated with perfluorooctyltriethoxysilane (Sigma-Aldrich, Oakville, ON, Canada) by vapor phase deposition. An accurate polymer copy of the wafer was obtained after double replication using polydimethylsiloxane (PDMS) and a UV-sensitive polyurethane as described in [14]. First, a 6 mm layer of 1:10 PDMS (Dow Corning, Corning, NY, USA) was poured on the wafer inside a Petri dish, followed by degassing under vacuum in a desiccator for 10 min. Next, the PDMS was cured in an oven for 24 h at 60 ˚C (VWR, Montreal, QC, Canada), then peeled from the wafer. To remove uncured monomers and other extractables, the PDMS replica was submersed in 70% ethanol for 24 h then baked at 60˚C for 4 h. Next, a drop of UV sensitive polyurethane (Norland Optical Adhesive 63 (NOA 63), Norland Products, Cranbury, NJ) was applied to the PDMS replica and cured by 600 W of UV light (Uvitron International, Inc., West Springfield, MA) for 30 s. The PDMS was then removed yielding an NOA replica of the original Si wafer pattern with 200 nm holes [14].

Nanocontact Printing

A flat PDMS stamp was cured against a perfluorooctyltriethoxysilane treated flat Si wafer. Following removal of the extractables as mentioned above, the flat PDMS stamp was inked with a 10 µL drop of phosphate buffered saline solution (PBS) containing 25 µg/mL of chicken immunoglobulin G (IgG) conjugated to Alexa Fluor 488 (Invitrogen, Burlington, ON, Canada). A plasma activated hydrophilic coverslip was then placed on the drop to spread the solution evenly across the surface of the hydrophobic PDMS stamp during a 5 min incubation period. After rinsing with PBS and double distilled water for 15 s each, the inked stamps were briefly dried under a stream of N_2 and immediately brought into contact with a plasma activated (PlasmaEtch PE-50, PlasmaEtch, Carson City, NV, USA) NOA master for 5 s. The PDMS and NOA were separated and proteins in the contact regions were transferred to the NOA. The remaining proteins on the PDMS were transferred to the final substrate by printing the PDMS stamp for 5 s onto a plasma activated glass coverslip.

Imaging and Analysis

Images of the original Si master were collected using scanning electron microscopy (SEM, JEOL, Japan). DNGs of fluorescent IgGs were imaged by fluorescence microscopy (TE2000 microscope, Nikon, Canada and CoolSNAP HQ^2 camera, Photometrics, USA). Images of the Si wafer were captured with a Panasonic Lumix GH3 DSLR equipped with an Olympus M. Zuiko Digital ED 60 mm macro lens. Dark field images were captured with an inspection microscope (LV150A microscope and Digital Sight DS-Fi1 camera, Nikon, Canada).

Results and Discussion

Ordered Gradient Algorithm

Ordered DNGs with continuously changing density were programmed by forming columns of nanodots with equal vertical spacing while varying the spacing between columns horizontally (Fig. 1A). The density of the gradient at any point along the length *l* is dictated by an input density function *D*, and is realized by placing a single nanodot into a virtual box to form a unit cell. The dimensions d_i of the unit cell at the i^{th} column of nanodots are given by the square root of the nanodot area A_{dot} divided by the density value at the given position (Eq. 1). In this algorithm, the size of the nanodot remains constant while the dimensions of the unit cell vary for each column of nanodots along the length.

$$d_i = \sqrt{\frac{A_{dot}}{D(l)}} \qquad (1)$$

The unit cell is largest at low densities and decreases at higher densities, matching the dimensions of the nanodot at a density equal to one. The unit cell dimensions for each column are calculated using an iterative approach, starting from the first column of nanodots at length zero. The position along the length for the next column of nanodots is given by the cumulative sum of unit cell dimensions (Fig. 1B).

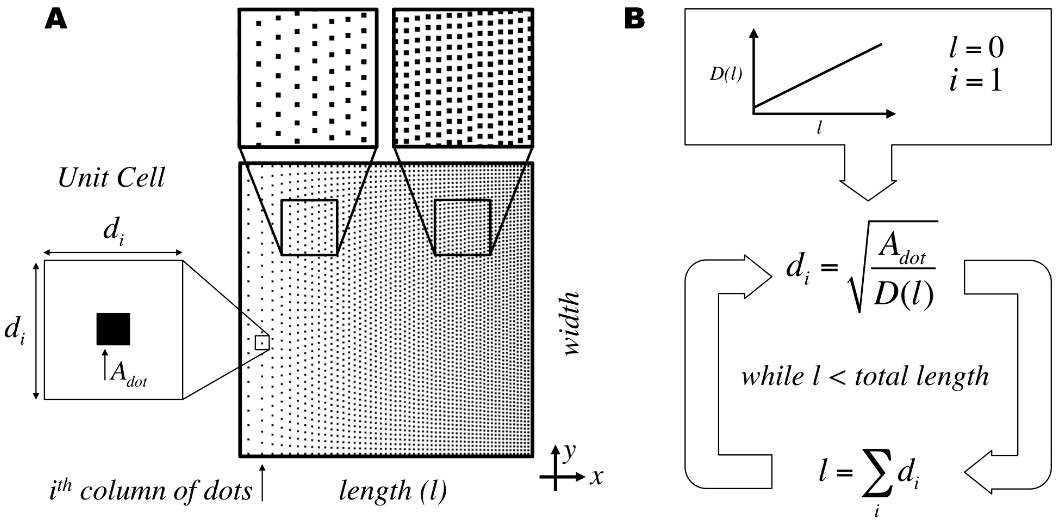


Figure 1: Ordered gradient algorithm for continuously changing density in Digital Nanodot Gradients (DNGs) using a unit cell approach. *(A) Schematic of a linear DNG and unit cell parameters for the i^{th} column of nanodots. (B) Calculation of unit cell dimensions at each position through an iterative process using the cumulative sum of unit cell dimensions to determine position along the length.*

The width of the gradient is divided by d_i to estimate the number of nanodots in the column. The value of d_i is then recalculated to equally space the integer number of nanodots along the width, ensuring that the distribution of nanodots is symmetrical. The unit cells are concatenated to form a column of points in the *y* direction with constant spacing as multiples of d_i. The *x* coordinate for each column is the position *l* at which the unit cell dimensions are calculated. Thus, using this algorithm it is easy to form a gradient with any slope as the size of the unit cell is directly derived from the value of the density function at a specific position. Fig. 2 shows a linear density curve spanning from a minimum density $D_{min}=0.01$ to a maximum density $D_{max}=0.30$ (Eq. 2) with corresponding unit cell dimensions along the length of the gradient. Eq. 2 is normalized to span the dynamic range over the total length $L=100\ \mu m$ of the gradient.

$$D(l) = (D_{\max} - D_{\min})\frac{l}{L} + D_{\min} \tag{2}$$

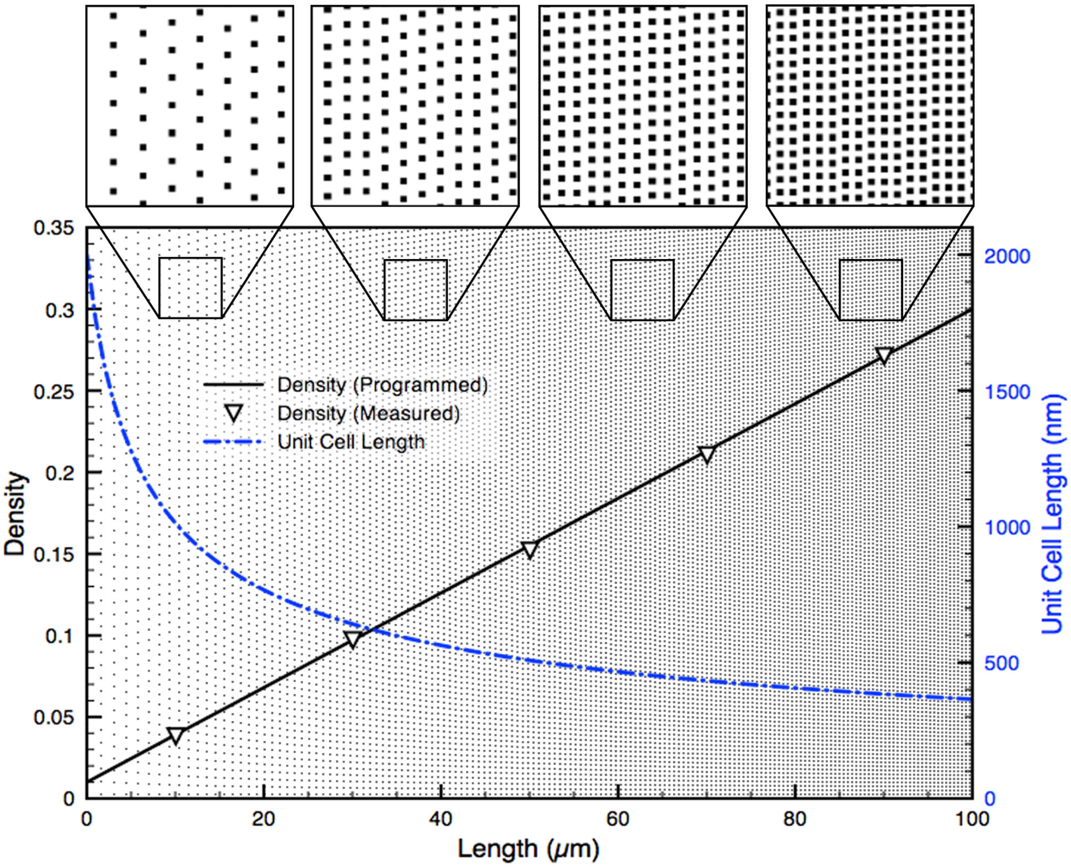


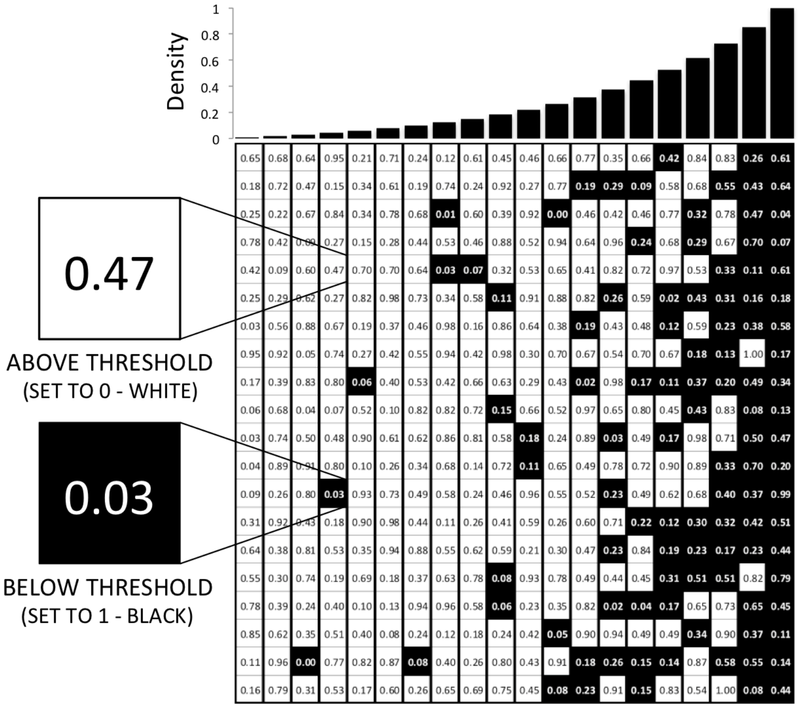
Figure 2: Programmed and measured density for an ordered linear DNG. *A 100 × 100 μm*^{2} linear DNG comprised of *0.04 μm*^{2} *nanodots and spanning from 0.01 to 0.30 density*. *The measured density (left axis) precisely matches the programmed density (R*^{2}*=0.9998). The unit cell dimensions (right axis) change with the inverse square root of the density. Top insets show various spacing of nanodots at positions along the DNG.*

To verify that ordered gradients matched the programmed density functions, gradients were exported as BMP images and the ratio of black to white pixels, averaged over several columns of nanodots, was used to measure density. The resulting gradients were found to match the programmed density functions with high fidelity (*R*^{2}*=0.9998*). Using this algorithm, a density of zero cannot be reached as it would require an infinitely large unit cell. Consequently, when designing gradients of fixed length, the lowest density that is accessible may be limited since large unit cells are required at very low density.

Random Gradient Algorithm

Naturally occurring gradients appear continuous at the microscale, but are in fact digital at the scale of molecules and proteins. The diffusion of biomolecules is subject to Brownian motion, and is therefore expected to be random. In an effort to mimic the nanoscale noise present with conditions *in vivo*, we present an approach to produce DNGs with fully randomized nanodot positions. Similar to ordered DNGs, random gradients have increasing density along the length and constant density along the width.

A tentative approach to generate DNGs with randomized positions was based on the conversion of a random matrix to a BMP image. A threshold dictated by an input density function was used to define if a nanodot is drawn on the BMP. The size of the array was such that each nanodot was represented by one pixel. Thus, for a 400 × 400 µm^2 sized area with 200 × 200 nm^2 nanodots, a 2000 × 2000 matrix with 4 million values is required. A matrix of uniformly distributed random numbers between 0 and 1 was generated using Matlab's pseudorandom number generator *'randn'*. Matrix values in each column were switched to either 0 or 1 based on the input density threshold, and the array was directly exported as a BMP image (Fig. 3). For example, with a threshold of 0.5, half the values in the column are expected to be black and the other half white.



0.65	0.68	0.64	0.95	0.21	0.71	0.24	0.12	0.61	0.45	0.46	0.66	0.77	0.35	0.66	0.42	0.84	0.83	0.26	0.61
0.18	0.72	0.47	0.15	0.34	0.61	0.19	0.74	0.24	0.92	0.27	0.77	0.19	0.29	0.09	0.58	0.68	0.55	0.43	0.64
0.25	0.22	0.67	0.84	0.34	0.78	0.68	0.01	0.60	0.39	0.92	0.00	0.46	0.42	0.46	0.77	0.32	0.78	0.47	0.04
0.78	0.42	0.09	0.27	0.15	0.28	0.44	0.53	0.46	0.88	0.52	0.94	0.64	0.96	0.24	0.68	0.29	0.67	0.70	0.07
0.42	0.09	0.60	0.47	0.70	0.70	0.64	0.03	0.07	0.32	0.53	0.65	0.41	0.82	0.72	0.97	0.53	0.33	0.11	0.61
0.25	0.29	0.62	0.27	0.82	0.98	0.73	0.34	0.58	0.11	0.91	0.88	0.82	0.26	0.59	0.02	0.43	0.31	0.16	0.18
0.03	0.56	0.88	0.67	0.19	0.37	0.46	0.98	0.16	0.86	0.64	0.38	0.19	0.43	0.48	0.12	0.59	0.23	0.38	0.58
0.95	0.92	0.05	0.74	0.27	0.42	0.55	0.94	0.42	0.98	0.30	0.70	0.67	0.54	0.70	0.67	0.18	0.13	1.00	0.17
0.17	0.39	0.83	0.80	0.06	0.40	0.53	0.42	0.66	0.63	0.29	0.43	0.02	0.98	0.17	0.11	0.37	0.20	0.49	0.34
0.06	0.68	0.04	0.07	0.52	0.10	0.82	0.82	0.72	0.15	0.66	0.52	0.97	0.65	0.80	0.45	0.43	0.83	0.08	0.13
0.03	0.74	0.50	0.48	0.90	0.61	0.62	0.86	0.81	0.58	0.18	0.24	0.89	0.03	0.49	0.17	0.98	0.71	0.50	0.47
0.04	0.89	0.91	0.80	0.10	0.26	0.34	0.68	0.14	0.72	0.11	0.65	0.49	0.78	0.72	0.90	0.89	0.33	0.70	0.20
0.09	0.26	0.80	0.03	0.93	0.73	0.49	0.58	0.24	0.46	0.96	0.55	0.52	0.23	0.49	0.62	0.68	0.40	0.37	0.99
0.31	0.92	0.43	0.18	0.90	0.98	0.44	0.11	0.26	0.41	0.59	0.26	0.60	0.71	0.22	0.12	0.30	0.32	0.42	0.51
0.64	0.38	0.81	0.53	0.35	0.94	0.88	0.55	0.62	0.59	0.21	0.30	0.47	0.23	0.84	0.19	0.23	0.17	0.23	0.44
0.55	0.30	0.74	0.19	0.69	0.18	0.37	0.63	0.78	0.08	0.93	0.78	0.49	0.44	0.45	0.31	0.51	0.51	0.82	0.79
0.78	0.39	0.24	0.40	0.10	0.13	0.94	0.96	0.58	0.06	0.23	0.35	0.82	0.02	0.04	0.17	0.65	0.73	0.65	0.45
0.85	0.62	0.35	0.51	0.40	0.08	0.24	0.12	0.18	0.24	0.42	0.05	0.90	0.94	0.49	0.49	0.34	0.90	0.37	0.11
0.11	0.96	0.00	0.77	0.82	0.87	0.08	0.40	0.26	0.80	0.43	0.91	0.18	0.26	0.15	0.14	0.87	0.58	0.55	0.14
0.16	0.79	0.31	0.53	0.17	0.60	0.26	0.65	0.69	0.75	0.45	0.08	0.23	0.91	0.15	0.83	0.54	1.00	0.08	0.44

Figure 3: Random gradients produced by a random matrix threshold approach. *A matrix of uniformly distributed random numbers is generated. Values greater than or equal to the density threshold are set to 0 (white), while values less than the threshold are set to 1 (black). The binary array can then be exported directly as a BMP image file.*

While this approach is straightforward and resulted in accurate matches between the programmed and measured density, it has a number of disadvantages. A large fraction of the information stored is uninformative (*i.e.* 0 values). To change the features from simple squares to shapes such as circles, the size of the array would have to be significantly increased, as multiple pixels are needed to represent non-square shapes. Most significantly, this approach does not provide a 'truly random' configuration since nanodots are aligned to a grid. Theoretically, it would be possible to further randomize the position of dots by subdividing the area further, *e.g.* using a 50 nm grid to position and draw 200 × 200 nm^2 pixels, but this would come at the cost of increased computational requirements and the possibility of overlap between adjacent dots.

Another approach considered for producing random gradients uses the coordinates obtained from the ordered gradient algorithm. The nanodots are randomly displaced from their original position within each unit cell by a given percentage. This approach eliminates possible overlap between neighboring nanodots, and allows for the degree of randomness to be controlled. Since overlap is avoided, this approach would mimic the self-repelling nature of proteins and discrete particles. However, this approach again constrains the nanodots to a grid, which does not accurately represent a random distribution.

To produce fully random gradients, we developed an alternative algorithm that uses equally sized boxes as previously used for ordered DNGs, but with a smaller size (1 × 400 μm^2) and random distribution of nanodots within each box. Owing to the random distribution within each box and to the small step-size, these gradients appear continuous at all scales. The number of nanodots within each box was calculated based on the density function at the box's position, and *x* and *y* coordinates were randomly generated for each nanodot. This approach stores only nanodot coordinates, thus reducing the file size and computation time significantly. The shape of the nanodots can be defined in the exported CIF file, allowing shapes such as circles, polygons, etc. to be used, adding to the flexibility of the approach. A consequence of the randomization of nanodot position is that overlap between nanodots is possible. The frequency of overlap increases with density, which in turn results in a lower density than expected based on the surface area of all nanodots. However, because the distribution of the nanodots is random, the overlap is predictable, and can be compensated for, so that the designed DNG matches the density function with high fidelity.

For each box with area A_{box}, we first compute the probability P_{cov} that any given point in the box is covered by a nanodot of area A_{dot} (Eq. 3).

$$P_{\text{cov}} = \frac{A_{dot}}{A_{box}} \tag{3}$$

The probability $P_{not\ cov}$ that this point will *not* be covered by the nanodot is simply $1 - P_{cov}$ (Eq. 4).

$$P_{\text{not cov}} = \left(1 - \frac{A_{dot}}{A_{box}}\right) = \frac{A_{box} - A_{dot}}{A_{box}} \tag{4}$$

The probability that this point will not be covered by *N* nanodots simultaneously can then be found (Eq. 5).

$$P_{\text{not cov}} = \left(\frac{A_{box} - A_{dot}}{A_{box}} \right)^{N} \quad (5)$$

To determine the total area of the box covered by nanodots (A_{cov}), the probability that a point *will* be covered is integrated over the area of the box (Eq. 6).

$$A_{\text{cov}} = \int_{0}^{A_{box}} (1 - P_{\text{not cov}})\, dA \quad (6)$$

$$A_{\text{cov}} = A_{box} - \frac{(A_{box} - A_{dot})^{N}}{A_{box}^{N-1}} \quad (7)$$

Eq. 7 can then be solved for N to determine the number of nanodots required for a given A_{cov} in each box.

$$N = \frac{\log(1 - A_{\text{cov}} / A_{box})}{\log(1 - A_{dot} / A_{box})} \quad (8)$$

Using Eq. 8 it is possible to obtain the number of nanodots required for any given density by substituting D for A_{cov}/A_{box}. As the area occupied approaches the area of the box, the number of nanodots required increases rapidly, and is infinite for a density of 1. The maximum density for a DNG was thus set to 0.9999. Without compensation, 9,999 nanodots of 200 × 200 nm^2 would be seeded into a 400 x 1 μm^2 box to reach this density. When accounting for the overlap 92,099 nanodots are required, roughly a tenfold increase.

We compared the programmed density with the measured density of a monotonic exponential curve with decay constant k (Eq. 9) using both the ordered and random gradient generation approaches. Eq. 9 is normalized to span the dynamic range over the length of the gradient. The random gradients produced with this algorithm were found to follow the programmed curve with high fidelity ($R^2 = 0.99986$), and accurately match the measured density of ordered gradients produced with the same input function (Fig. 4).

$$D(l) = (D_{\max} - D_{\min}) \frac{e^{k\frac{l}{L}} - 1}{e^{k} - 1} + D_{\min} \quad (9)$$

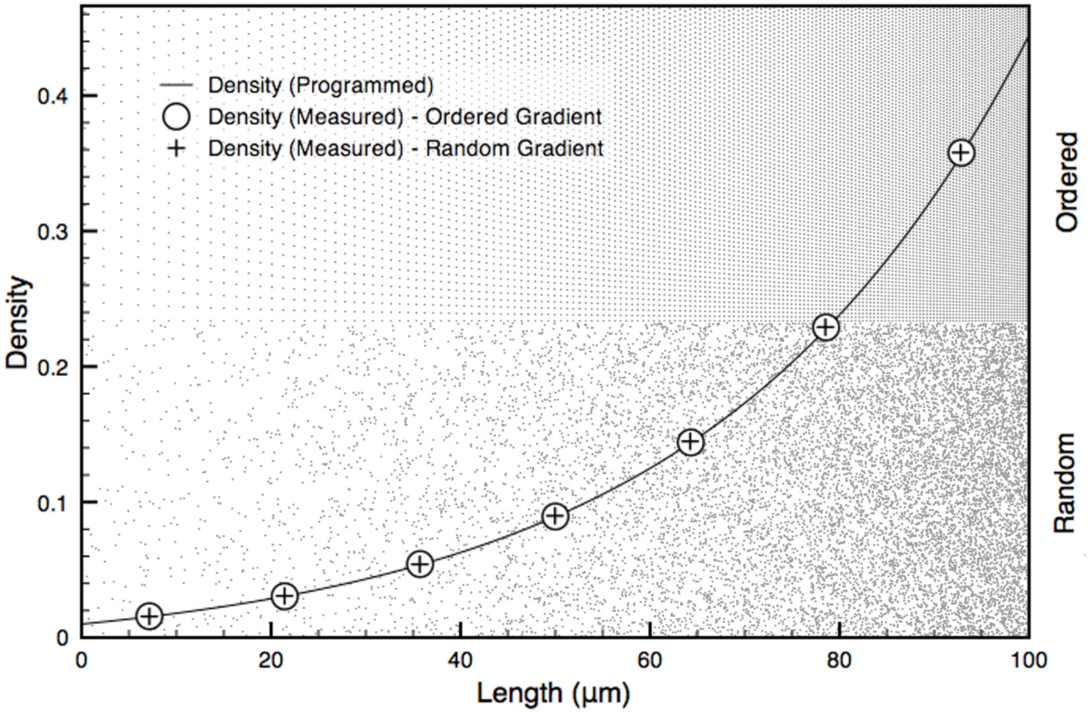


Figure 4: Ordered and random DNGs superposed with their programmed exponential density function and actual, measured density. *A 100 × 100 µm^2 exponential (k=3) DNG spanning from a density of 0.01 to 0.30* is shown in ordered (top half) and random (bottom half) form. The random DNG was programmed by subdividing it into 1 µm wide boxes with random seeding of nanodots and compensating for overlap. *The measured density follows the programmed exponential curve with high fidelity for both ordered (R^2=0.99996) and random (R^2=0.99986) DNGs.*

The distribution of neighboring nanodots can be parsed to verify whether it satisfies the conditions of randomness. Since density changes along the length of the DNG, randomness can only be assessed in the direction of constant density, perpendicular to the gradient. The randomness along each box was verified using Ripley's K function based on the number of nanodots N_{pi} within a distance s from the i^{th} point p_i taken over the sum of all points n and normalized by the area λ (Eq. 10).

$$K(s) = \frac{1}{n}\sum_{i=1}^{n} N_{p_i}(s)/\lambda \qquad (10)$$

For a homogenous, random Poisson distribution, the expected value of Eq. 10 is πs^2. Deviations from πs^2 indicate regions of clustering or dispersion [24]. Using the coordinates of nanodots from the DNGs, *K(s)* was found to lie within a 95% confidence interval obtained from 10 simulations of randomly distributed coordinates at the same density. This suggests the nanodots are spatially homogenous and randomly distributed compared to ordered gradients, which lie outside the confidence interval (Fig. 5).

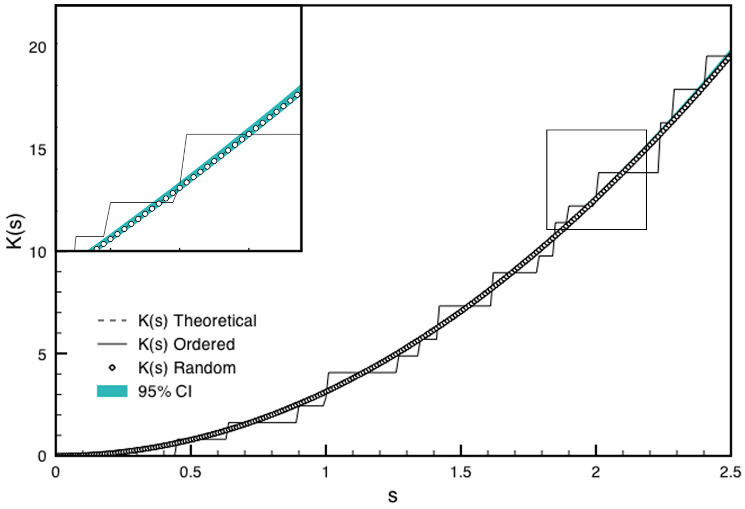


Figure 5: Ripley's K analysis for randomness and spatial homogeneity of a sample box produced with the ordered and random DNG algorithms. *K(s) for ordered and random gradients for a 10 × 400 μm^2 box at 0.20 density are shown. A 95% confidence interval was calculated as 1.96 times the standard deviation of K(s) from 10 simulations of randomly distributed points at 0.20 density. The inset shows a magnified portion of the graph indicated by the box.*

Non-monotonic gradients

We have shown that linear and exponential gradients can be produced using either the ordered or random algorithms. These curves are monotonic, meaning they only ever increase or decrease. Given the approaches outlined here, more complex gradients can be easily generated from any input density function, specifically non-monotonous curves. To study the ability of cells to recognize an average gradient in a non-monotonous environment, we propose linear and exponential gradients that are superposed with sinusoidal functions with either constant amplitude, or with linear or exponentially changing amplitude. In the anticipation of future cell experiments, we also programmed a series of sinusoids with varying frequency and amplitude, while having an average slope of zero to act as negative controls.

Here, we demonstrate the flexibility of the algorithms with the production of non-monotonic gradients. One such gradient produced is a sinusoid with exponentially increasing amplitude superposed an exponential gradient (Fig. 6). Eq. 11 gives the input density function for such a curve, where A is the amplitude, B is the number of oscillations, and k_1 and k_2 are the decay constants for the average gradient and amplitude of the sinusoid, respectively.

$$D(l) = (D_{\max} - D_{\min})\frac{e^{k_1\frac{l}{L}} - 1}{e^{k_1} - 1} + A\sin\left(2\pi B\frac{l}{L}\right)\frac{e^{k_2\frac{l}{L}} - 1}{e^{k_2} - 1} + D_{\min} \qquad (11)$$

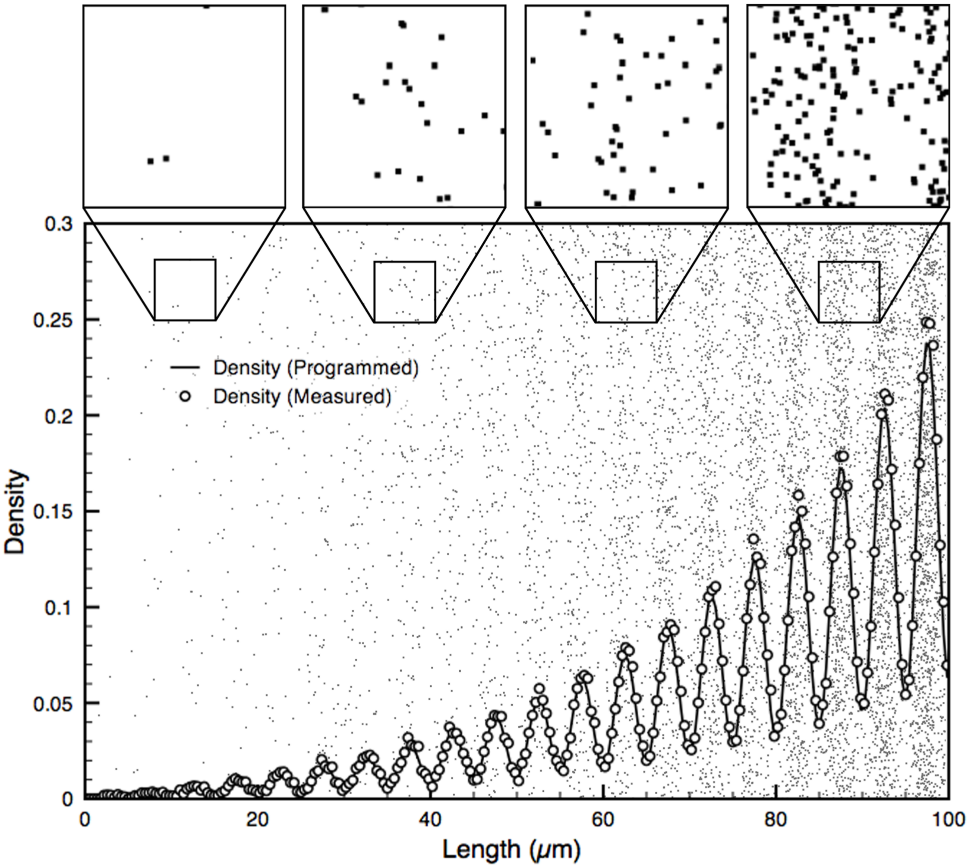


Figure 6. Non-monotonic random DNG superposed with its density function and actual, measured density. *A 100 x 100 μm^2 non-monotonic DNG. The programmed density function (line) is a sinusoid (A=0.10, B=20) with an average exponential trend (k_1=3) and exponentially increasing amplitude (k_2=3). Measured density from the BMP (circles) accurately follows the programmed function (R^2=0.9900). Insets show close-up views of nanodots and reveal increasing overlap of nanodots at higher densities.*

One-Hundred-Gradient Array

The flexibility of the gradient algorithms and the fabrication method discussed below was leveraged by producing an array of 100 distinct gradients within a 35 mm^2 area (Fig. 7, Table. S1). The array is comprised of 20 ordered and 80 random gradients, with densities ranging from 0.0002 to 0.4444 for a maximum dynamic range of 3.85 OM. For ordered gradients, this corresponds to a maximum pitch of 14.8 μm at low density, and 100 nm at high density. The minimum density is limited by the size of the cell and its capacity to sense the gradient, *i.e.* the cell will not sense the gradient if dot spacing exceeds one third the size of the cell, which we previously assessed using the migratory response of C2C12 myoblasts [14]. The maximum density is defined by the resolution of e-beam lithography, which limits the minimum pitch to 100 nm. To address how randomness affects cell navigation, 20 gradients (10 linear and 10 exponential) were produced as both ordered (#1-20) and random (#81-100) gradients. This portion of the array will serve to address gradient sensing mechanics for cell migration that may arise from either (i) the absolute concentration of the gradient at a given location or (ii) the concentration ratio between the cell's leading and trailing edges.

The other 60 gradients are non-monotonic and random. These consist of 14 sinusoids with no average slope (#21-34) that serve as controls. These vary in frequency and amplitude, and may have constant or increasing amplitude along their length. Controls for ordered gradients were designed in our prior work. The remaining 46 gradients (#35-80) have sinusoidal curves with various levels of complexity. These include sinusoids with linearly (#41-60) and exponentially (#61-80) increasing average density, and non-monotonic functions with linearly increasing amplitude and average density (#35-36) and exponentially increasing amplitude and average density (#37-40) to demonstrate the flexibility of this new approach in gradient generation. As discussed, the sinusoidal curves with different amplitude and frequency may introduce obstacles for cells trying to sense the overarching gradient, and replicate effects that may occur due to cell "mosaicism" in a controlled manner.

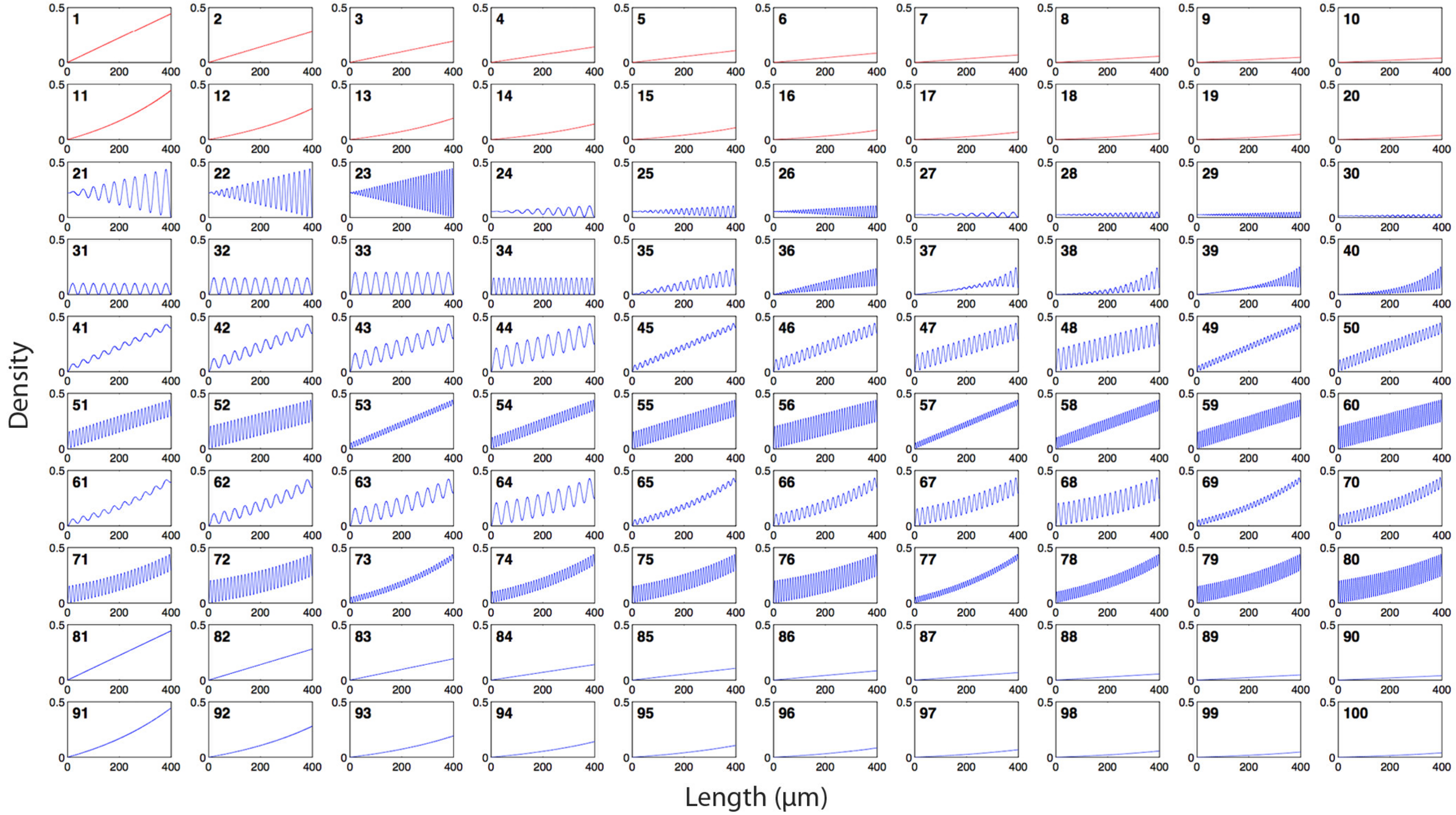


Figure 7: Density functions of the array of 100 DNGs. *Each box shows the density function of one gradient. Functions 1-20 were produced with the ordered gradient algorithm (red), and functions 21-100 were produced with the random gradient algorithm (blue). Functions 1-10 and 81-90 are linear. Functions 11-20 and 91-100 are exponential. Functions 21-34 are sinusoidal with no slope (controls), where 21-30 have linearly increasing amplitude and 31-34 have constant amplitude. Functions 35-36 feature a linear slope superposed with a sinusoid that has linearly increasing amplitude. Functions 37-40 feature an exponential slope superposed with a sinusoid that has exponentially increasing amplitude. Functions 41-60 have a constant slope superposed with a sinusoid of constant amplitude. Functions 61-80 are exponential slopes superposed with sinusoids of constant amplitude.*

Gradient Array Fabrication

The hundred-gradient array was etched 100 nm deep into a Si wafer by e-beam lithography (Fig. 8). The integrity of individual dots for ordered and random gradients was confirmed by scanning electron microscopy (SEM) (Fig. 9).

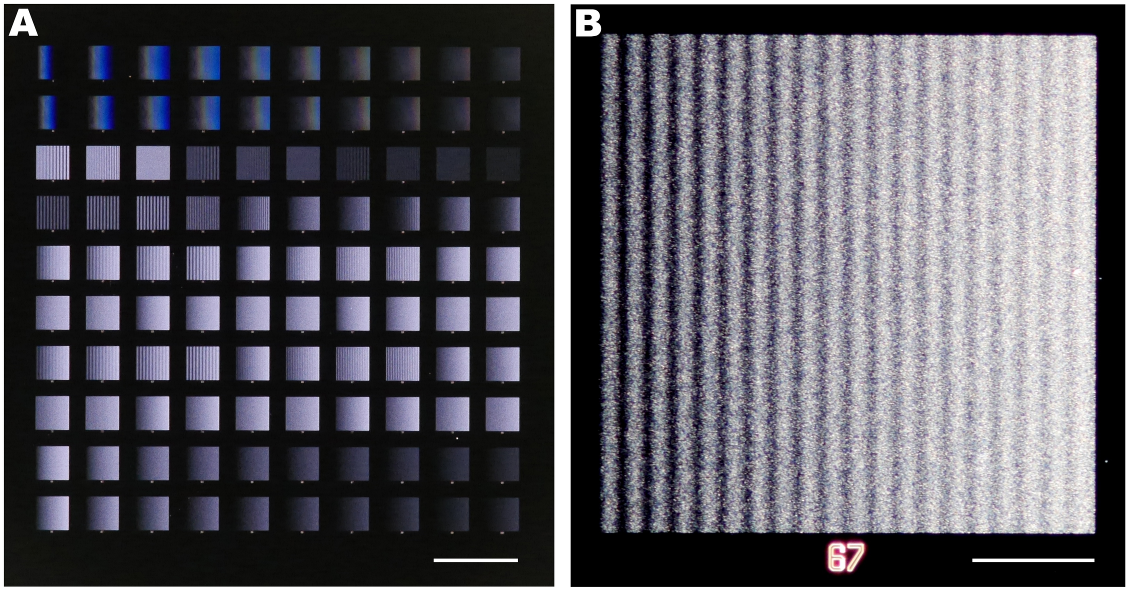


Figure 8: Optical images of the 100-gradient-array. *The 100 DNGs were patterned into a Si wafer using e-beam lithography. (A) Image of all gradients; scale bar is 1 mm. (B) Dark-field image of DNG 67 which is a random sinusoid with exponentially increasing average density; scale bar is 100 µm.*

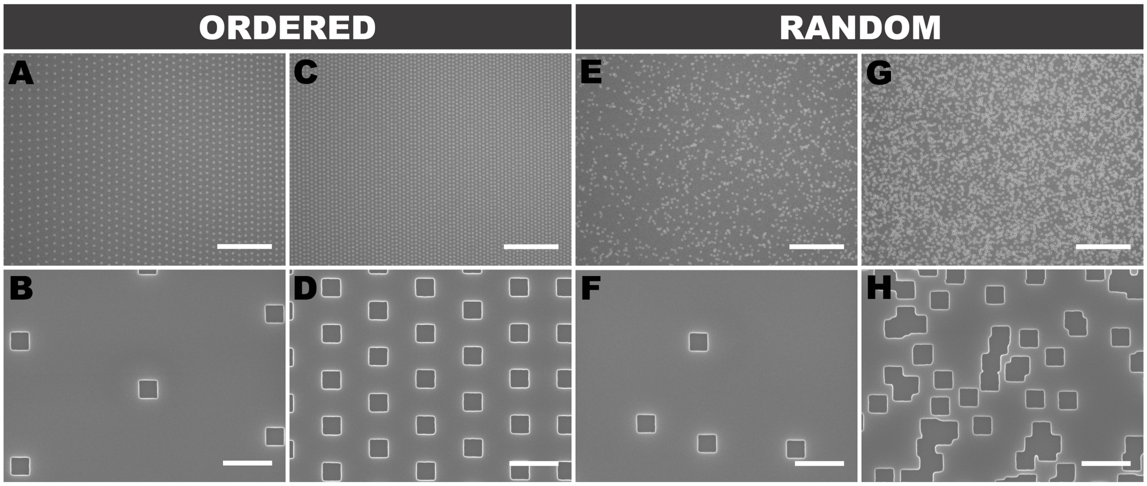


Figure 9: Nanodot distribution in ordered and random DNGs. *SEMs of ordered (A-D) and random (E-H) exponential gradients at low (A,B,E,F) and high (C,D,G,H) densities. (H) highlights the random overlap between nanodots that is compensated for by the random DNG algorithm. Scale bars are 10 um (top row) and 500 nm (bottom row).*

To translate the etched Si wafer into substrate-bound protein gradients, we employed lift-off nanocontact printing [14]. First, a PDMS intermediate replica was produced, followed by a second replication into low-cost Norland Optical Adhesive. A flat PDMS stamp was inked with a protein solution, and through contact with the plasma-activated NOA lift-off stamp, a monolayer of protein was selectively removed from the surface of the flat stamp leaving behind the digital nanodot protein pattern. Next, the flat PDMS stamp was brought into contact with a plasma-activated glass slide to transfer the DNG pattern. To confirm the accuracy of the replication and printing process, images of the design and the printed protein DNG were digitally overlapped and compared (Fig. 10), revealing a high fidelity between the two patterns.

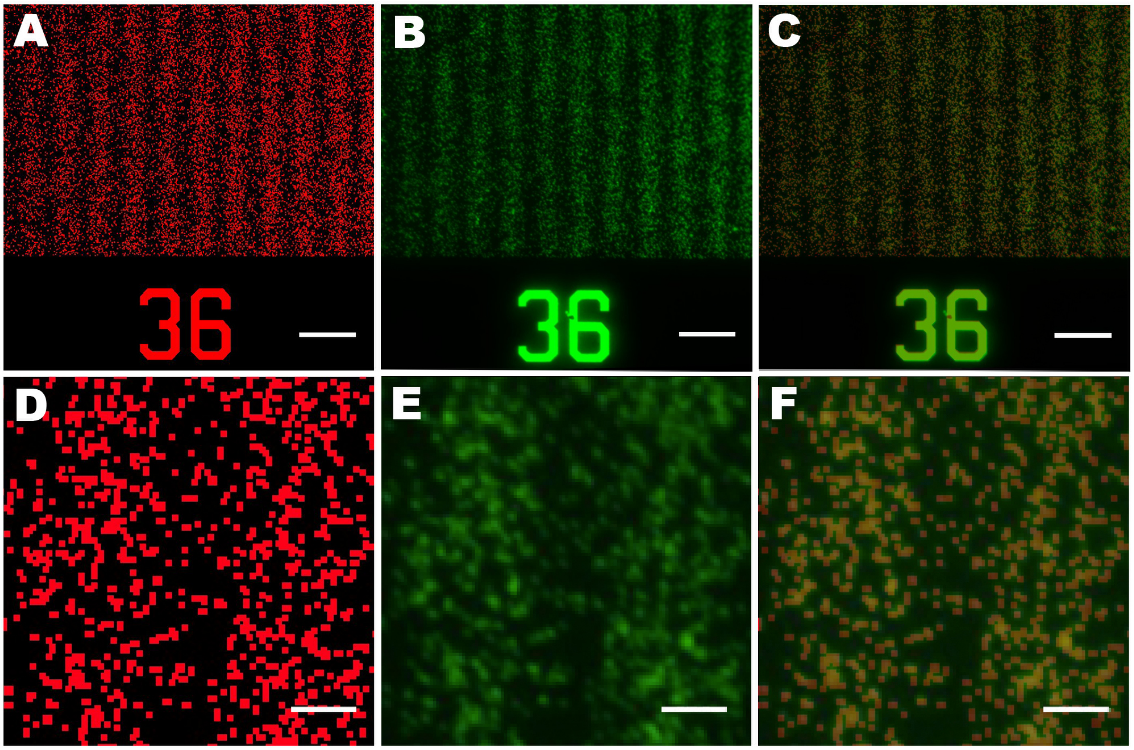


Figure 10: The printed nanodot pattern accurately replicates the design. *Bitmap image of the programmed design colored in red (A, D) compared with a fluorescent image of nanocontact printed IgG by lift-off (B, E, green) and merged (C, F). In the inset (E), out of ~1000 spots, 21 are missing, indicating that the replication works well. Scale bars A-C are 10 μm, scale bars D-F are 2 μm.*

The pattern overlay in Fig. 10 was characterized by thresholding the printed image with boundaries of 31 and 255 in ImageJ and comparing it to the bitmap image (Fig. S1). The total number of nanodots is difficult to determine owing to the random overlap, but we estimate that there are ~1000 spots, and using the above threshold, 21 were not found in the print. Overall, the transfer process is thus accurate to ~98% in this example. Dust particles on the Si wafer or on the intermediate PDMS replica, or mechanical damage due to the replication process could account for the absence of these protein dots.

Whereas the fidelity of the replication and printing process is thus high, it might be improved further by employing more durable polymers during the replication process as well as by working in a cleanroom environment throughout.

Conclusion

Patterned substrate-bound protein gradients are to a valuable tool to study a number of biological processes such as neuronal development or regeneration. The novel algorithms presented here can provide control over geometry and noise in DNGs. Combined with high-throughput patterning, an array of one hundred DNGs with linear, exponential, and non-monotonic slopes featuring 57 million spots over an area of 35 mm^2 can be patterned at once in a matter of minutes. The diversity of DNGs shown here will help study and quantify the mechanisms by which cells sense and navigate through immobilized gradients. There are many opportunities for refining digital nanodot patterns. Firstly, to mimic the self-repellent nature of proteins adsorbing to surfaces, it might be useful to develop an algorithm that prevents, or limits, the overlap of nanodots, and thus optimally fills the available space. Secondly, whereas here we opted to overlap sinusoidal curves in one direction, such curves might be generated in two directions to better mimic local accumulation of guidance cues as puncta. Thirdly, it might be possible to program density functions that introduce clusters of noise that more accurately replicate the noise and mosaicism imposed by individual cells *in vivo*. Fourthly, it should be possible to pattern overlapping DNGs of different proteins that run in the same, or different directions, as well as produce any type of navigational landscapes that are found *in vivo* simply by converting the recorded densities into digital nanodot patterns following the trend of rapid prototyping of replicas of living tissues [25]. Before expanding the nanodot patterning, it will be important to test and validate the current DNGs and establish the optimal conditions along with the suitable non-patterned reference surfaces for each study [26].

Acknowledgments

We acknowledge Nabila Zaman for her help with gradient algorithms, and Marta Garnelo-Abellanas and Veronique Laforte for proofreading this article.

References

1. McLaughlin T, O'Leary DDM (2005) Molecular gradients and development of retinotopic maps. Annu Rev Neurosci 28: 327-355.
2. Chung BG, Flanagan LA, Rhee SW, Schwartz PH, Lee AP, et al. (2005) Human neural stem cell growth and differentiation in a gradient-generating microfluidic device. Lab on a Chip 5: 401-406.
3. Goodman CS (1996) Mechanisms and molecules that control growth cone guidance. Annual Review of Neuroscience 19: 341-377.
4. Kim BK, Lee JW, Park PJ, Shin YS, Lee WY, et al. (2009) The multiplex bead array approach to identifying serum biomarkers associated with breast cancer. Breast Cancer Res 11: R22.
5. Lander AD, Nie Q, Wan FYM (2002) Do morphogen gradients arise by diffusion? Developmental Biology 247: 471-471.
6. Gurdon JB, Bourillot PY (2001) Morphogen gradient interpretation. Nature 413: 797-803.
7. Keenan TM, Folch A (2008) Biomolecular gradients in cell culture systems. Lab on a Chip 8: 34-57.
8. Joanne Wang C, Li X, Lin B, Shim S, Ming G-l, et al. (2008) A microfluidics-based turning assay reveals complex growth cone responses to integrated gradients of substrate-bound ECM molecules and diffusible guidance cues. Lab on a Chip 8: 227-237.
9. Mai J, Fok L, Gao HF, Zhang X, Poo MM (2009) Axon Initiation and Growth Cone Turning on Bound Protein Gradients. Journal of Neuroscience 29: 7450-7458.
10. Dertinger SKW, Jiang XY, Li ZY, Murthy VN, Whitesides GM (2002) Gradients of substrate-bound laminin orient axonal specification of neurons. Proceedings of the National Academy of Sciences of the United States of America 99: 12542-12547.
11. Von Philipsborn AC, Lang S, Bernard A, Loeschinger J, David C, et al. (2006) Microcontact printing of axon guidance molecules for generation of graded patterns. Nature Protocols 1: 1322-1328.
12. Coyer SR, García AJ, Delamarche E (2007) Facile Preparation of Complex Protein Architectures with Sub-100-nm Resolution on Surfaces. Angewandte Chemie International Edition 46: 6837-6840.
13. von Philipsborn AC, Lang S, Loeschinger J, Bernard A, David C, et al. (2006) Growth cone navigation in substrate-bound ephrin gradients. Development 133: 2487-2495.
14. Ricoult SG, Pla-Roca M, Safavieh R, Lopez-Ayon GM, Grütter P, et al. (2013) Large Dynamic Range Digital Nanodot Gradients of Biomolecules Made by Low-Cost Nanocontact Printing for Cell Haptotaxis. Small 9: 3308-3313.
15. Raser JM, O'Shea EK (2005) Noise in gene expression: Origins, consequences, and control. Science 309: 2010-2013.

16. Dalby MJ, Gadegaard N, Tare R, Andar A, Riehle MO, et al. (2007) The control of human mesenchymal cell differentiation using nanoscale symmetry and disorder. Nature Materials 6: 997-1003.
17. Kantawong F, Burgess KEV, Jayawardena K, Hart A, Burchmore RJ, et al. (2009) Whole proteome analysis of osteoprogenitor differentiation induced by disordered nanotopography and mediated by ERK signalling. Biomaterials 30: 4723-4731.
18. Tonazzini I, Meucci S, Faraci P, Beltram F, Cecchini M (2013) Neuronal differentiation on anisotropic substrates and the influence of nanotopographical noise on neurite contact guidance. Biomaterials 34: 6027-6036.
19. Dickinson R, Tranquillo R (1993) A stochastic model for adhesion-mediated cell random motility and haptotaxis. Journal of Mathematical Biology 31: 563-600.
20. Smith JT, Elkin JT, Reichert WM (2006) Directed cell migration on fibronectin gradients: Effect of gradient slope. Experimental Cell Research 312: 2424-2432.
21. Weber M, Hauschild R, Schwarz J, Moussion C, de Vries I, et al. (2013) Interstitial Dendritic Cell Guidance by Haptotactic Chemokine Gradients. Science 339: 328-332.
22. Sweetser DA, Birkenmeier EH, Hoppe PC, McKeel DW, Gordon JI (1988) Mechanisms underlying generation of gradients in gene expression within the intestine: an analysis using transgenic mice containing fatty acid binding protein-human growth hormone fusion genes. Genes & Development 2: 1318-1332.
23. Ming GL, Wong ST, Henley J, Yuan XB, Song HJ, et al. (2002) Adaptation in the chemotactic guidance of nerve growth cones. Nature 417: 411-418.
24. Kiskowski MA, Hancock JF, Kenworthy AK (2009) On the Use of Ripley's K-Function and Its Derivatives to Analyze Domain Size. Biophysical journal 97: 1095-1103.
25. Derby B (2012) Printing and Prototyping of Tissues and Scaffolds. Science 338: 921-926.
26. Ricoult SG, Thompson-Steckel G, Correia JP, Kennedy TE, Juncker D (2014) Tuning cell–surface affinity to direct cell specific responses to patterned proteins. Biomaterials 35: 727-736.

Supplementary Material

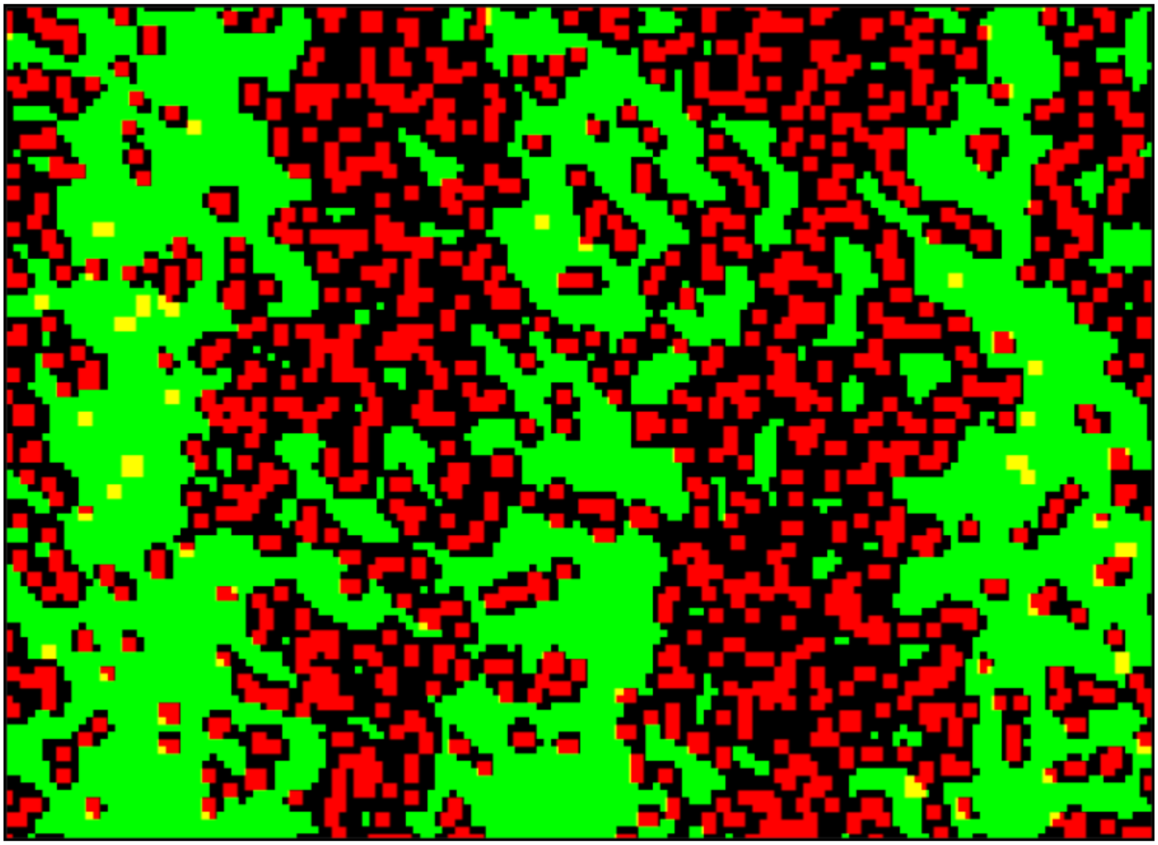


Figure S1: Image processing procedure to assess alignment of the DNG design and print. *The fluorescent image of nanocontact printed IgG was first thresholded in ImageJ with boundary values of 31 and 255. The image was then transformed to binary and the binary values inverted to facilitate visualization. The edited fluorescent image (green) was then merged with the BMP (red) and yellow dots, indicative of non-printed dots, were counted to determine to what extent the print matched the design.*

Supplementary Table 1

#	function	O/R	min (%)	max (%)
1	linear	O	0.02	44.44
2	linear	O	0.02	28.03
3	linear	O	0.02	19.27
4	linear	O	0.02	14.06
5	linear	O	0.02	10.71
6	linear	O	0.02	8.43
7	linear	O	0.02	6.81
8	linear	O	0.02	5.61
9	linear	O	0.02	4.70
10	linear	O	0.02	4.00
11	exponential	O	0.02	44.44
12	exponential	O	0.02	28.03
13	exponential	O	0.02	19.27
14	exponential	O	0.02	14.06
15	exponential	O	0.02	10.71
16	exponential	O	0.02	8.43
17	exponential	O	0.02	6.81
18	exponential	O	0.02	5.61
19	exponential	O	0.02	4.70
20	exponential	O	0.02	4.00
21	sin_linamp	R	0.02	43.34
22	sin_linamp	R	0.02	43.89
23	sin_linamp	R	0.02	44.17
24	sin_linamp	R	0.02	10.44
25	sin_linamp	R	0.02	10.58
26	sin_linamp	R	0.02	10.64
27	sin_linamp	R	0.02	4.82
28	sin_linamp	R	0.02	4.88
29	sin_linamp	R	0.02	4.91
30	sin_linamp	R	0.02	2.74
31	sinusoidal	R	0.02	10.02
32	sinusoidal	R	0.02	15.02
33	sinusoidal	R	0.02	20.02
34	sinusoidal	R	0.02	15.02
35	sin_pluslin_linamp	R	0.02	23.25
36	sin_pluslin_linamp	R	0.02	23.69
37	sin_plusexp_expamp	R	0.02	24.21
38	sin_plusexp_expamp	R	0.02	24.04
39	sin_plusexp_expamp	R	0.02	25.08
40	sin_plusexp_expamp	R	0.02	24.99
41	sin_pluslin	R	0.02	42.55
42	sin_pluslin	R	0.02	42.75
43	sin_pluslin	R	0.02	42.99
44	sin_pluslin	R	0.02	43.23
45	sin_pluslin	R	0.02	43.48
46	sin_pluslin	R	0.02	43.59
47	sin_pluslin	R	0.02	43.71
48	sin_pluslin	R	0.02	43.83
49	sin_pluslin	R	0.02	43.80
50	sin_pluslin	R	0.02	43.87
51	sin_pluslin	R	0.02	43.96
52	sin_pluslin	R	0.02	44.04
53	sin_pluslin	R	0.02	43.96
54	sin_pluslin	R	0.02	44.01
55	sin_pluslin	R	0.02	44.08
56	sin_pluslin	R	0.02	44.14
57	sin_pluslin	R	0.02	44.05
58	sin_pluslin	R	0.02	44.10
59	sin_pluslin	R	0.02	44.15
60	sin_pluslin	R	0.02	44.20
61	sin_plusexp	R	0.02	41.58
62	sin_plusexp	R	0.02	41.86
63	sin_plusexp	R	0.02	42.21
64	sin_plusexp	R	0.02	42.58
65	sin_plusexp	R	0.02	42.95
66	sin_plusexp	R	0.02	43.12
67	sin_plusexp	R	0.02	43.30
68	sin_plusexp	R	0.02	43.49
69	sin_plusexp	R	0.02	43.43
70	sin_plusexp	R	0.02	43.55
71	sin_plusexp	R	0.02	43.68
72	sin_plusexp	R	0.02	43.81
73	sin_plusexp	R	0.02	43.68
74	sin_plusexp	R	0.02	43.77
75	sin_plusexp	R	0.02	43.87
76	sin_plusexp	R	0.02	43.96
77	sin_plusexp	R	0.02	43.83
78	sin_plusexp	R	0.02	43.90
79	sin_plusexp	R	0.02	43.98
80	sin_plusexp	R	0.02	44.06
81	linear	R	0.02	44.44
82	linear	R	0.02	28.03
83	linear	R	0.02	19.27
84	linear	R	0.02	14.06
85	linear	R	0.02	10.71
86	linear	R	0.02	8.43
87	linear	R	0.02	6.81
88	linear	R	0.02	5.61
89	linear	R	0.02	4.70
90	linear	R	0.02	4.00
91	exponential	R	0.02	44.44
92	exponential	R	0.02	28.03
93	exponential	R	0.02	19.27
94	exponential	R	0.02	14.06
95	exponential	R	0.02	10.71
96	exponential	R	0.02	8.43
97	exponential	R	0.02	6.81
98	exponential	R	0.02	5.61
99	exponential	R	0.02	4.70
100	exponential	R	0.02	4.00

Legend

O/R	Ordered/Random
sin_linamp or sin_expamp	Sinusoid with linearly or exponentially increasing amplitude
sin_pluslin or sin_plusexp	Sinusoid plus a linear or exponential average trend
sin_pluslin_linamp	Sinusoid with linearly increasing amplitude and linear average trend
sin_plusexp_expamp	Sinusoid with exponentially increasing amplitude and exponential average trend