www.sciencemag.org/cgi/content/full/316/5832/1753/DC1

Supporting Online Material for

Local Signaling Networks That Regulate Cell Morphology Defined by Quantitative Morphological Signatures

Chris Bakal,* John Aach,* George Church, Norbert Perrimon

*To whom correspondence should be addressed.
E-mail: cbakal@receptor.med.harvard.edu (C.B.); aach@receptor.med.harvard.edu (J.A.)

Published 22 June, *Science* **316**, 1753 (2007)
DOI: 10.1126/science.1140324

This PDF file includes:

TABLE OF CONTENTS

Index of Figures

Index of Tables

Supplemental Materials and Methods

Protocols

Overexpression constructs and dsRNA: All RFP-tagged constructs were created using Gateway Technology (Invitrogen) by subcloning of cDNAs into the pPRW (N-terminal RFP, UASp promoter) or pPWR (C-terminal RFP, UASp promoter) destination plasmids (Drosophila Genome Resource Center). Table S1 describes these constructs in detail. cDNAs were kind gifts from Greg Bashaw (University of Pennsylvania), Rick Cerione (Cornell University), Ulrike Gaul (Rockefeller University), Chihiro Hama (University of Tokyo), and Bingwei Lu (Stanford University). Alternatively, cDNAs were PCR amplified from full-length *Drosophila* ORFs provided by Drosophila Genome Resource Center (Berkeley, USA). dsRNA was prepared as described in detail at www.flyrnai.org.

Cell culturing and stochastic labeling: *Drosophila* DM-BG2 cells (referred to as BG-2 cells in this paper) were cultured in Shields and Sang M3 insect media (Sigma), 10% Fetal Bovine Serum, 40 μg/ml (Sigma), 10 μg/ml Insulin (Sigma), and Penicillin-Streptomycin (Gibco). All cells were transfected with *actin*-GAL4, and *UAS*-GFP containing plasmids using Effectene transfection reagent (Qiagen). For dsRNA experiments, cells were co-transfected with dsRNAs as described in detail at www.flyrnai.org. For overexpression experiments, cells were co-transfected with plasmids encoding RFP-tagged proteins.

Treatment Conditions: As Rho signaling has been extensively implicated in the regulation of the cytoskeleton, we explicitly sought to generate a number QMSes corresponding to a diverse spectrum of Rho, Rac, or Cdc42 activity. We proposed that these signatures would not only represent distinct cellular morphologies, but that other cellular states with similar signatures could be classified as playing a role in Rho-, Rac-, or Cdc42-specific signaling pathways. In order to generate different types of GTPase activity, we overexpressed constitutively activated GTP-locked mutants of *Drosophila* Rho1 (RhoV14), Rac1 (RacV12), and Cdc42 (Cdc42V12), “fast-cycling” mutants of Rho (Rho30L) and Rac (RacF28L), a “slow-cycling” mutant of Cdc42 (Cdc42Y32A), as well as full-length and N-terminally truncated forms of particular *Drosophila* RhoGEFs. Furthermore, we specifically targeted the majority of *Drosophila* RhoGEFs, RhoGAPs, and GTPases for dsRNA-mediated gene silencing. GTP-locked forms of both Rho and Rac have been long observed to stimulate dramatic changes in the actin cytoskeleton and can profoundly affect cell morphology (*2, 3*). Similar to GTP-locked mutants, “fast/slow” cycling mutants of GTPases are also hyperactivated enzymes, but due to the fact they cycle through both GDP- and GTP- bound states, are significantly more biologically potent. For example, while overexpression of GTP-locked forms does not induce transformation in mammalian cells, fast/slow-cycling mutants are highly oncogenic (*4-6*). N-terminal truncation has repeatedly been shown to stimulate RhoGEF activity, which is likely due to the autoinhibitory effects of regions N- terminal to the catalytic DH/PH domains (*7*).

Our final dataset comprises 249 treatment conditions (TCs) corresponding to: (1) The overexpression by transient transfection of 20 different RFP-tagged mutant forms of Rho GTPases, RhoGEFs, kinases, and other regulators of the microtubule and actin cytoskeletons (see Table S1). (2) 173 dsRNAs chosen at random from a larger collection of dsRNAs targeting all known GTPases, GEF, GAPs, and other genes implicated in cytoskeletal organization. This collection of dsRNA overlaps considerably with the collection of ~900 dsRNAs used by our lab in previous morphological screens (*8*). (3) An additional 45 dsRNAs targeting the majority of known *Drosophila* RhoGEFs, GAPs, and GTPases (4) Overexpression of an activated form of the RhoGEF SIF/still-life in combination with various dsRNAs chosen at random.

Overexpression Construct	**Mutation**	**Consequence of Mutation**	**Reference**
ΔN-CG3799	Deletion of 517 N-terminal amino acids of Drosophila CG3799 (isoform A).	Predicted to be constitutively activated.	This study
ΔN-RhoGEF3	Deletion of 245 N-terminal amino acids of Drosophila RhoGEF3 (isoform C)	Likely not constitutively active (*9*).	This study
ΔN-SIF	Deletion of 1214 N-terminal amino acids of Drosophila SIF (Type 2).	Predicted to be constitutively activated as per previously described mutants with similar truncations (*10*).	This study
Aurora-B kinase (human) constitutively active	Mutation in kinase domain	Hyperactivated kinase	This study
CG3799 full-length	N/A	N/A	This study
Cdc42Y32A (Human)	Y32A	Promotes “slow-cycling” between GTP- and GDP- bound states of Cdc42 resulting in hyperactivation of Cdc42	(*6*)
dLis1 full-length	N/A	N/A	This study
dMEMO. Full-length CG8031. Drosophila ortholog of mammalian Memo (*11*).	N/A	N/A	This study
dPar-1 full-length	N/A	N/A	(*12*)
dSTRAD. Full-length CG7693. Drosophila ortholog of mammalian STRAD (*13*).			This study
Gαι65A full-length	N/A	N/A	This study
GEF64C full-length	N/A	N/A	(*14*)
Moody-beta full-length	N/A	N/A	(*15*)
Neuroglian (*Drosophila)* full-length	N/A	N/A	This study
RacF28L (Human)	F28L	Promotes fast-cycling between GDP- and GTP- bound states of Rac resulting in hyperactivation of Rac	(*5*)
RacV12	G12V	Decreases intrinsic GTPase activity and causes Rac to be unresponsive to RacGAPs.	(*3*)
RhoF30L (Human)	F30L	Promotes fast-cycling between GDP- and GTP- bound states of Rho resulting in hyperactivation of Rho	(*5*)
RhoV14	G14V	Decreases intrinsic GTPase activity and causes Rho to be unresponsive to RhoGAPs.	(*2, 16*)
SIF full-length (Type 2)	N/A	N/A	(*10*)
TumL/JAK	Mutation in kinase domain	Results in hyperactivation of *Drosophila* JAK kinase	(*17*)

Table S1: Summary of RFP-tagged expression constructs used in this study.

Image acquisition: Following transfection of BG-2 cells with plasmids and/or RNAi, cells were cultured in 384-well plates and fixed in 4% paraformaldehyde in PBS 4 days post-transfection. Images were acquired using an automated Nikon TE300 microscope with a 40× objective and HTS MetaMorph software (Universal Imaging) running an automated Mac5000-driven stage, filter wheel and shutter (Ludl Electronic Products), an automated Pifoc focusing motor (Piezo) and an Orca-ER cooled-coupled device camera (Hamamatsu). For the majority of Treatment Conditions (TCs) involving a single dsRNA (213 dsRNAs), images were acquired in semi-automated and blinded fashion from a single well. The identity of these dsRNA was determined following the completion of segmentation, feature extraction, and QMS-based clustering procedures. For the remainder of the TCs involving single dsRNAs, images were acquired from multiple (2-12) wells from the same 384-well plate. In cases where cells were transfected with RFP-tagged overexpression constructs, images were acquired from multiple wells from the same 384-well plate. As a control, two GFP-alone TCs were imaged at the beginning (November 2005) and completion (October 2006) of the experiments described in this study.

Rho activation assay: Rho activity in whole cell-lysates was determined using the G-LISA RhoA Activation Assay Biochem Kit (Cytoskeleton Inc.). The assay was performed according to manufacturer's instructions.

Cell Image Selection Software

Stochastic labeling (see **Protocols** above and main text) very successfully diminished the density of labeled cells in each image to the point where individual cells were easily distinguished by eye. We tested several automated image segmentation algorithms and found that each still yielded frequent instances of multiple cells combined into single segments, cells divided into multiple segments, and inaccurate segment boundaries. Factors contributing to error generation included: (i) the uneven distribution of label intensities within labeled cells (i.e., some cells were brightly and others dimly labeled), (ii) background, (iii) the very irregular shape of BG-2 cells, (iv) the inability to use other non-stochastically labeled image channels to assist segmentation. Instead of completely automating segmentation, we developed a software application (CellSegmenter) for computer-assisted segmentation.

CellSegmenter is a MatLab GUI application that allows a user to choose and display a TIFF image of a set of cells, manually adjust an image intensity threshold until the threshold boundary best fits a cell boundary, and then to select this thresholded cell boundary as a cell segment using point and click operations. Different thresholds may be specified for different cells in the image field, and cells that are in contact may be separated manually by drawing short CellSegmenter "borders" between them. When cell segment specification is complete, the segmentation may be saved for subsequent processing by image analysis algorithms or for subsequent re-adjustment of segment boundaries by CellSegmenter. Because CellSegmenter requires human intervention, it is not suitable for very high-throughput applications. However, it is appropriate for small-to-medium throughput applications involving up to a few 1000s of images in each of which up to 10s of cell segments are selected.

CellSegmenter requires MatLab 7.1 or higher and was written, tested, and used on Windows XP computers exclusively and we cannot provide assurances that it will run on other versions or operating systems. However, the CellSegmenter source code is available on http://arep.med.harvard.edu/QMS/ along with documentation on the installation and usage of CellSegmenter.

Feature Analysis

Overview

Over the course of ~10 months, 12,601 individual cell segments were generated using CellSegmenter. Automated image analysis algorithms were developed to compute 145 mathematical values (features) for each of these segments from the cell segment image created by CellSegmenter and the original GFP intensity image. While information on these cells derived from other stains and labels was obtained and used in some cases (e.g., to confirm co-transfection and expression of RFP-tagged constructs in GFP expressing cells), no such information was used in the calculation of features. The features were designed to interrogate aspects of the overall geometry and size of the cell segments, the stochastic GFP label intensity, and the statistical distribution and 'texture' of this intensity with relation to cell geometry. Finally, many features measured attributes of the shape of the cell boundary as rendered by the cell segment, including the number, size, shape, and distribution of processes and undulations of the boundary as analyzed at both a small and a large scale. These overall geometry and boundary-level features represent information unobtainable from complex cell images obtained without stochastic labeling because of cell crowding and overlaps normally prevent their clear discernment. The feature set also included a number of previously published features reported to be useful for analyzing the cytoskeletal behavior of cells.

On a feature by feature basis, features obtained for each cell segment were normalized to Z scores relative to their values over a subset of 145 GFP control cells which were transfected only with a construct coding for constitutive expression of GFP without other perturbation. All subsequent analysis used these normalized features.

The 12,601 cell segments were obtained in 14 separate batches and corresponded to 273 treatment classes (TCs), where a TC represents a cell sample perturbed by a set of dsRNAs and/or constructs driving constitutive expression of a wild-type or mutant gene. In most cases, a TC where a dsRNA is used to inhibit a target gene was derived from cell segments from a single well of a single plate; however a subset of 32 TCs were derived from cells in multiple wells. These were analyzed separately to test for replicability of results (see **Clustering and Replicability Analysis** in **Quantitative Morphological Analysis**, below) and were subsequently aggregated. Outside of these 32 TC replicants, a second set of GFP control segments (October 2006) was also generated outside of the initial sample of GFP 145 control cells (November 2005). Because only the original 145 GFP controls were used to normalize all other cells, and extensive computations had already been done to develop neural network classifiers (see below) for several TCs based on these normalized values, the second set of GFP control segments was *not* aggregated with the first, but was kept as a separate TC. TCs where cells are overexpressing RFP-tagged proteins are typically segments from cells in multiple wells that were cultured and fixed in parallel.

Normalized feature data for all 12,601 cell segments, and TC means and standard deviations of all these data, are provided as supplemental data files on our web site http://arep.med.harvard.edu/QMS.

In cases where the same gene was targeted by different amplicons, or the same amplicon was present in multiple wells, the individual TCs were merged into a single TC. With the mergings taken into account, the dataset described in this study contained 249 TCs.

Feature Generation Process

The feature generation process is described in Figure S1. Cell samples are arrayed in a subset of wells in 384-well plates in which each well contains distinct dsRNAs and/or other constructs, so that each well corresponds to a different RNAi or overexpression treatment. Each well also contains a GFP construct that stochastically labels the cells (see **Protocols** and main text), so that only a random, sparse, and dispersed subset of cells within the crowded population appears in the GFP channel in a fluorescent microscope field. Under the hypothesis that competent cells will pick up all constructs in the well, GFP-labeled cells will also express the perturbation specified by the dsRNA or other constructs in the well.

TIFF images of fluorescent microscopy fields were acquired as described in **Protocols** above. As any given image field contains multiple labeled cells, the images are segmented using the CellSegmenter application (see **Cell Image Selection Software** above). CellSegmenter enables a user to adjust intensity thresholds until they match visible boundaries of the cell, draw small borders between touching cells, and select and save cell segments considered to be good representations of well imaged cells. For any given TIFF image, the output of CellSegmenter operations is a single csf file that describes all of the segments selected from the image, and a set of csm files each of which stores the CellSegmenter-defined boundary of a single cell processed and selected from the image (see Figure S1a).

In the next step (Figure S1b), the csf, csm, and original TIFF image for each cell segment are read by a MatLab program designed to generate a large number (154) of distinct numerical features describing attributes of the segment. The csm file provides information on the boundary defined for the cell segment, the TIFF image contains the intensity of the label captured by fluorescence microscopy, and the csf file indicates the location of the csm-captured cell segment boundary within the larger TIFF image. The csm file is therefore the source of all features that provide information on the shape of the cell, while the csm and the intensity information in the TIFF image together are the

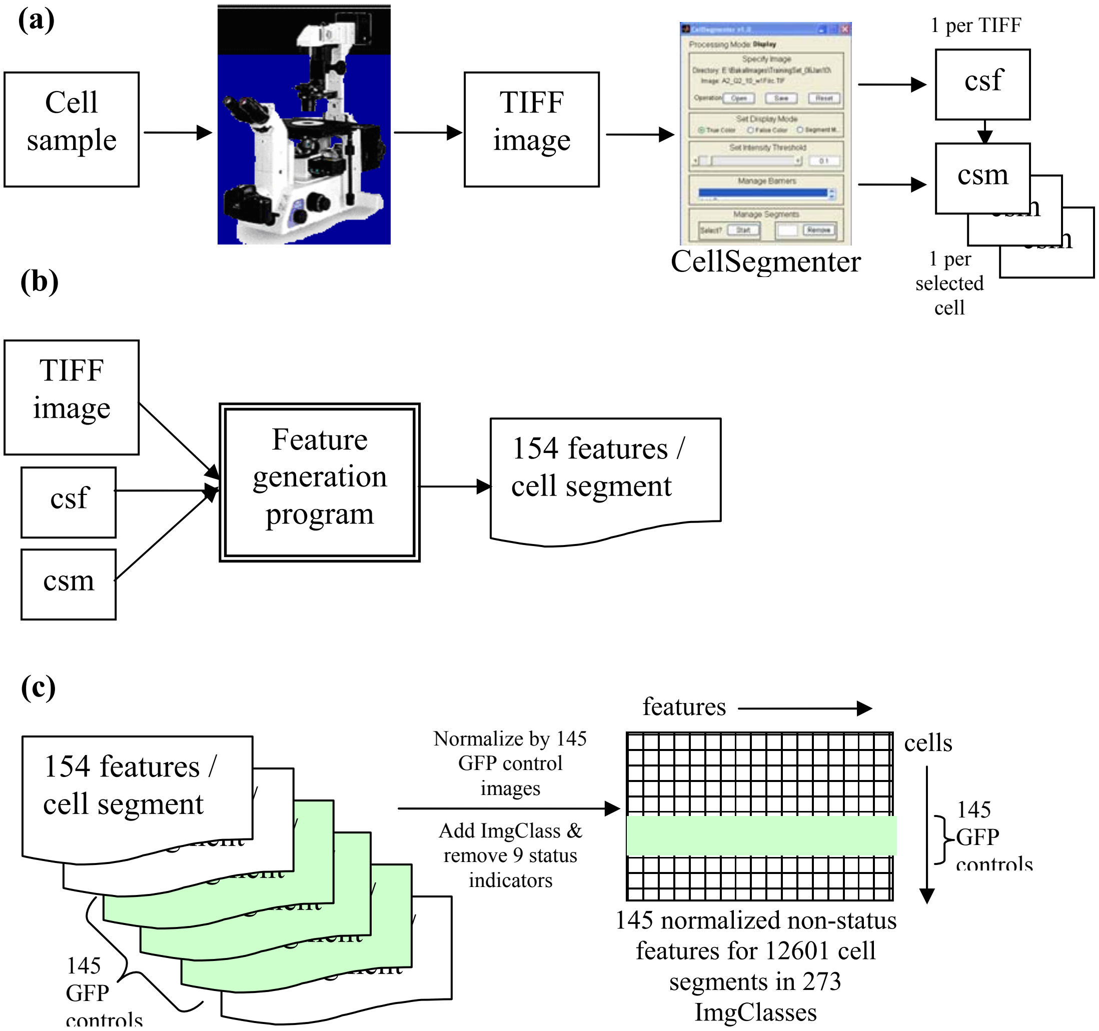

(a)
Cell sample
TIFF image
CellSegmenter
1 per TIFF
csf
csm
1 per selected cell
(b)
TIFF image
csf
csm
Feature generation program
154 features / cell segment
(c)
154 features / cell segment
145 GFP controls
Normalize by 145 GFP control images
Add ImgClass & remove 9 status indicators
features
cells
145 GFP controls
145 normalized non-status features for 12601 cell segments in 273 ImgClasses

Figure S1: Feature generation. (a) Fluorescence microscopy of cell samples generates TIFF images that are analyzed with CellSegmenter to yield cell segments that represent selected individual cells in image. Segment positions and boundaries are saved in csf and csm files. (b) For each cell segment, csf, csm, and original TIFF files are analyzed by a feature generation program to produce 154 numerical features / segment. (c) Among the cell segments for which features are computed are 145 GFP control cells (green) that serve as a reference set. Features for *all* segments are normalized to Z scores for each segment feature relative to the values in this reference set. Nine status indicator features are removed, and each segment is annotated with an ImgClass that describes the treatment (an RNAi or overexpression) used to generate it.

source of all features that describe the intensity distribution of the cell. All features derive only from the GFP signal generated in the stochastically labeled cells. While in theory additional labels such as DAPI or phalloidin may be used to acquire information about other cell consitutents, such labels will be global rather than stochastic and yield signal for *all* cells in the crowded image rather than just the sparse GFP-stochastically labeled cells; thus it will not be possible to distinguish what part of the signal from these labels is associated with an isolated GFP-labeled cell from the part of the signal associated with non-GFP-labeled cells lying above or below it. The 154 features are described in the section **Definitions of Individual Features** below.

The 154 features computed by the feature generation program were obtained for 12,601 individual cell segments comprising 273 distinct treatments processed in 14 different batches over the course of ~10 months. Mathematically, the different features contain different kinds of information and have values that lie on many different scales. To ease feature comparison and analysis, the 154 features were therefore normalized with reference to a set of 145 cells from the second batch of cells that were set up as GFP controls (ImgClass = gfp1) (see Figure S1c). These 145 cells were treated only with the GFP construct for stochastic labeling and no other dsRNA or expression construct. The normalized value of any feature of any cell segment is simply the Z score of the un-normalized feature value relative to the mean and standard deviation of the un-normalized feature values of the 145 GFP controls. At this time, nine 'status indicator' features (see below) were removed, leaving 145 normalized non-status feature values per cell segment, and each cell segment was annotated with an ImgClass that describes its treatment class. A subset of 32 of the 273 treatment classes comprised cells from multiple (2-5) wells, some of which were processed in different batches, and which therefore comprise biological replicates. In the final version of the file of normalized feature values for all segments, all segments in replicate treatment classes are given the same ImgClass and combined in computing means, variances, and other statistics for the ImgClass. However, in one series of calculations described below, the individual well cell segments from these 32 treatment classes were *not* combined in order to test the consistency of the feature values obtained in replicate treatments (see **Clustering and Replicability Analysis**). The 32 treatment classes that comprise replicates are given in Table S2.

In addition to these 32 treatment classes comprising replicate samples, a second sample of 29 GFP images without additional dsRNA or overexpression constructs was analyzed in batch 14. Although these comprise a replicate of gfp1 GFP control treatment class described above, they were held apart as a separate treatment class (ImgClass = gfp_06Oct17) and therefore *not* combined with gfp1 segments in normalizing features or in computing statistics for GFP controls. The reason was that the gfp1 class alone was used in normalizing the data on which neural network classifiers were trained and optimized (see manuscript): therefore, to avoid confusion about which GFP control cells were used in classifier training, and likewise avoid the high computational overhead entailed by combining the new and old GFP controls and retraining the neural networks, the gfp_06Oct17 images were left apart the earlier set of gfp1 controls.

Generation of features for all 12,601 cell segments over all 14 batches of images was performed on a single cluster of computers to minimize the possibility that different versions of MatLab running on different systems could compute some MatLab built-in functions differently. This was observed once during early testing. Over the course of the 10 months during which the cells in this study were analyzed, there were two changes

ImgClass	#reps
CdGAPr:CdGAPr_RNAi__P1I2	2
cenG1A:P1O17__P1P17	2
CG10188:CG10188_RNAi__P1D11	2
CG11490:P1B1__P1B3	2
CG12102:P1N16__P1I12	2
CG15611:P1M10__P1P19	2
CG30115:CG30115_RNAi__P1N21	2
CG30158:P1K6__P1M6	2
CG30372:P1D17__P1N6	2
CG30440:CG30440_RNAi__P1J8	2
CG30456:P1L10__P1O18	2
CG3799:CG3799_RNAi__P1G10	2
CG8243:P1I14__P1O6	2
empty:P1F9__P1I23	2
GEF64C:GEF64C_v361__GEF64C_08319__GEF64C_08318	3
G-gamma30A:P1N15__P1M15	2
Graf:Graf_RNAi__P1P7	2
jitterbug:P1F13__P1I4	2
mbc:mbc_16995__mbc_36492	2
paxillin:P1M19__P1C11	2
pbl:pbl_33336__pbl_11381__pbl_26301__pbl_RNAi__pbl_33335	5
RacGAP50C:RacGAP50C_33345__RacGAP50C_07575	2
Rho1:Rho1_RNAi__P1F21__P1J16	3
RhoGAP15B:RhoGAP15B_RNAi__P1M9	2
RhoGAP16F:RhoGAP16F_RNAi__P1I11	2
RhoGAPp190:RhoGAPp190_RNAi__P1O9	2
RhoGEF2:RhoGEF2_07531__RhoGEF2_29373	2
RhoGEF3:P1O16__P1E2	2
RhoGEF4:P1F6__RhoGEF4_11011	2
Sar1:P1M14__P1E20	2
Sos:Sos_RNAi__P1N17	2
Trio:Trio_RNAi__P1B4	2

Table S2: Treatment classes (ImgClasses) comprising multiple samples (replicates)

in MatLab release. Judged by small scale test recalculations of data, we saw no evidence of significant change in MatLab function calculation.

Image Normalization

All images are single-channel grayscale images of the GFP label that were normalized so that the maximum intensity in the image is 1.0. The simple computer-assisted thresholding used to segment images that is enabled by the CellSegmenter application (see **Cell Image Selection Software**) precludes the need for elaborate background analysis and segment filtering, and most features are computed directly from the normalized intensity image without background subtraction. Exceptions arise for computation of the *GFP bright spot* (see below) and the *segment mass image*.

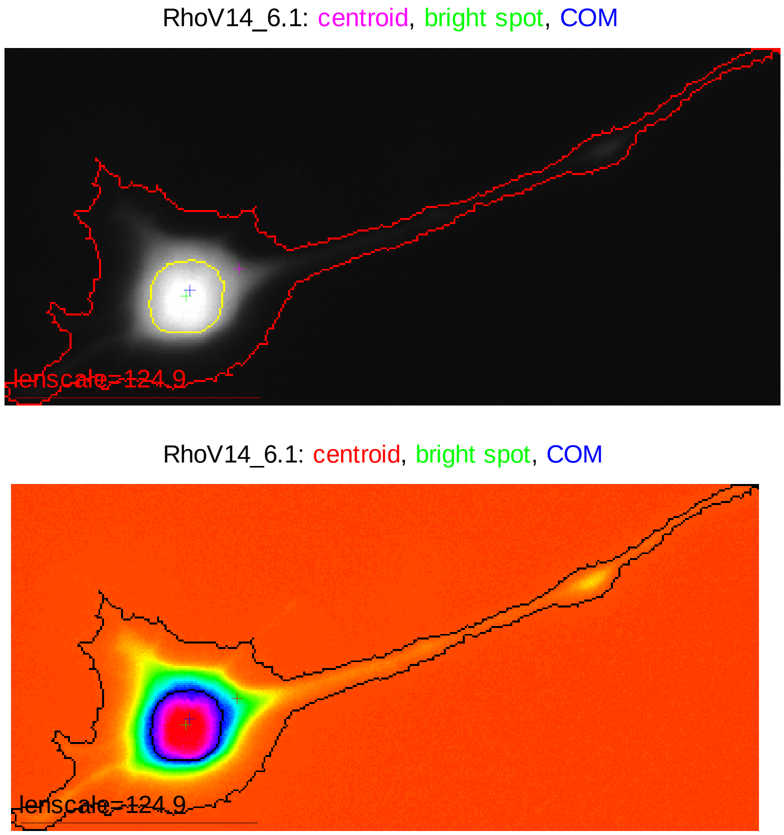

RhoV14_6.1: centroid, bright spot, COM
lenscale=124.9
RhoV14_6.1: centroid, bright spot, COM
lenscale=124.9

Figure S2: Cell RhoV14_6.1 with boundary defined by CellSegmenter, plus the GFP bright spot boundary and its three cell centers: *GFP centroid* (centroid), *GFP bright spot centroid* (bright spot) and *center of mass* (COM). Top: cell grayscale image with centers identified by colors in figure title. Bottom: cell false color image with same centers and boundaries indicated in black. lenscale = *length_scale* (see text) indicates the scale.

The *segment mass image* is used for features which interpret the pixel intensity distribution as a probability distribution. It is derived from the normalized intensity image by:

1. subtracting the value of the threshold used to define the segment in CellSegmenter
2. setting the value of any pixels exterior to the segment to 0
3. setting the value of any pixel interior to the segment that is <0 to 0
4. normalizing so that the total sum of all pixel values is 1.

Note that 3 above implies that it is possible for a pixel within a CellSegmenter-defined segment boundary to have an intensity value < the value of the threshold that defines the segment. This situation only arises because CellSegmenter fills all holes in threshold-defined segments before returning a segment boundary. Normally, when applying a 'raw' threshold to a cell image (especially one with a complex 'texture') the resulting raw segment may contain holes that represent intensity depressions within the cell that are deep enough to go below the threshold. By filling these holes, CellSegmenter represents

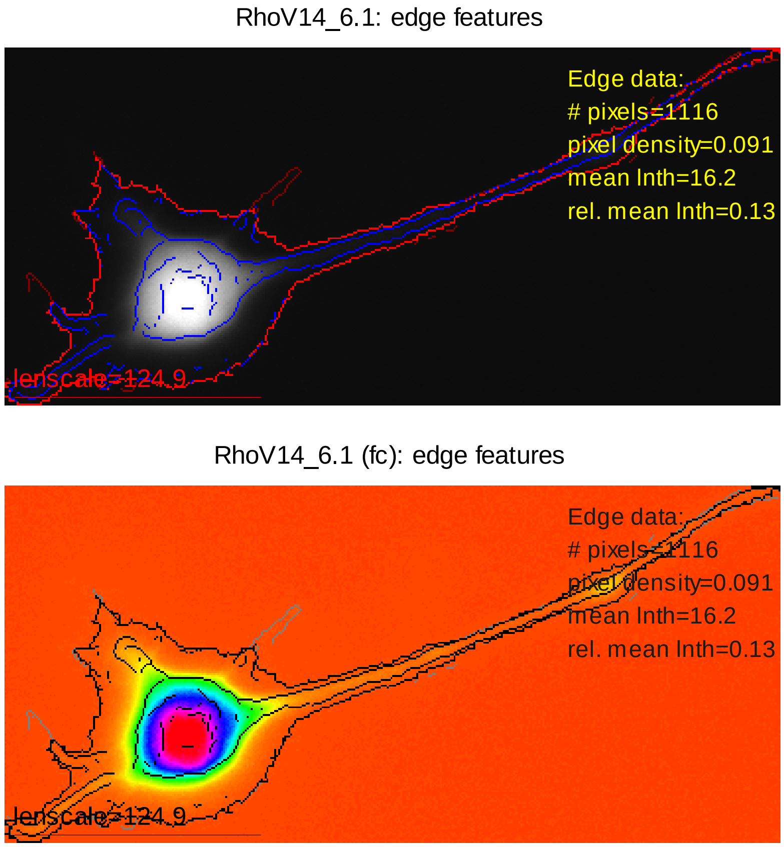
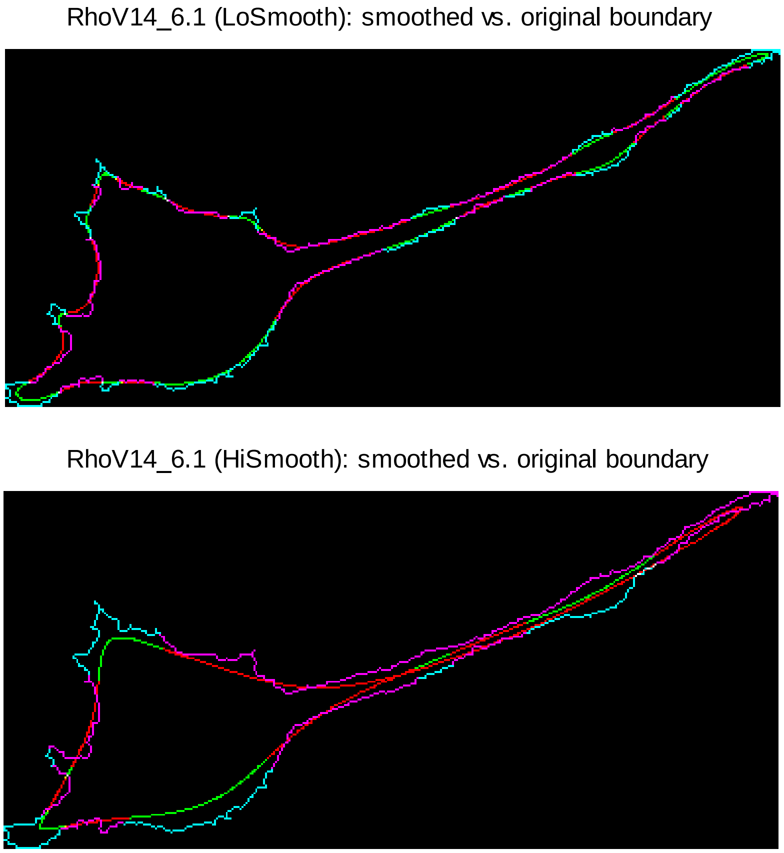


Figure S3: Edge image of cell RhoV14_6.1 with several edge feature values indicated. Edges interior to the cell segment are indicated by blue lines in grayscale image (top) and by black lines in false color image (bottom). Edges that may exist outside of or extend beyond the cell segment are indicated with dim lines and are not considered in edge feature calculations. Edge features calculated include the total number of edge pixels, the pixel density (total number of edge pixels / cell segment area), the mean edge length, and the mean edge length divided by *length_scale* (bottom left).

the cell as a region with a simple closed boundary that ignores these interior depressions. Thus, subtracting the CellSegmenter threshold intensity from the original pixel intensities in this region (step 1) will yield negative values (step 3) in any such hole.

Figures Illustrating Feature Analysis

Feature generation for cell segments involves computing numerical values from the geometry of the segment and the intensity distribution within it. To illustrate the aspects of the geometry and intensity distribution that are analyzed, Figures S2-S12 are presented for a particular cell RhoV14_6.1 = cell segment 1 from the image RhoV14_6, a sample that was treated with the GFP stochastic label construct and a construct expressing a constitutively active Rho (RhoV14). This cell happens to be the cell in the training set for the RhoV14 neural network classifier (see **Quantitative Morphological Analysis** and main text) that scored highest on this classifier.



RhoV14_6.1 (LoSmooth): smoothed vs. original boundary

RhoV14_6.1 (HiSmooth): smoothed vs. original boundary

Figure S4: Boundary smoothing for cell RhoV14_6.1: LoSmooth (top) and HiSmooth (bottom) (see text). In each case, the smoothed boundary is represented by a contour which alternates color between green and red, with green indicating arcs of positive curvature and red indicating arcs of negative curvature. The original cell boundary (see also Figure S2) is indicated as an alternating purple and cyan contour, with cyan for regions of original boundary assigned green in the smoothed boundary and purple for regions of original boundary assigned red in the smoothed boundary. Note how smoothing simplifies the original convoluted boundary by removing small irregular protrusions, and that high smoothing achieves a larger degree of simplification and shape abstraction than low smoothing.

General Aspects of Features

1 *Status features:* Nine of the 154 features generated for each cell segment are status / quality indicators that describe the success or quality of various aspects of segmentation, feature analysis, and processing. These nine features are described here even though they are removed from the normalized non-status feature data (see above).

2 *Key reference elements:* Many features are computed with respect to reference elements within the cell segment. Key reference elements include:

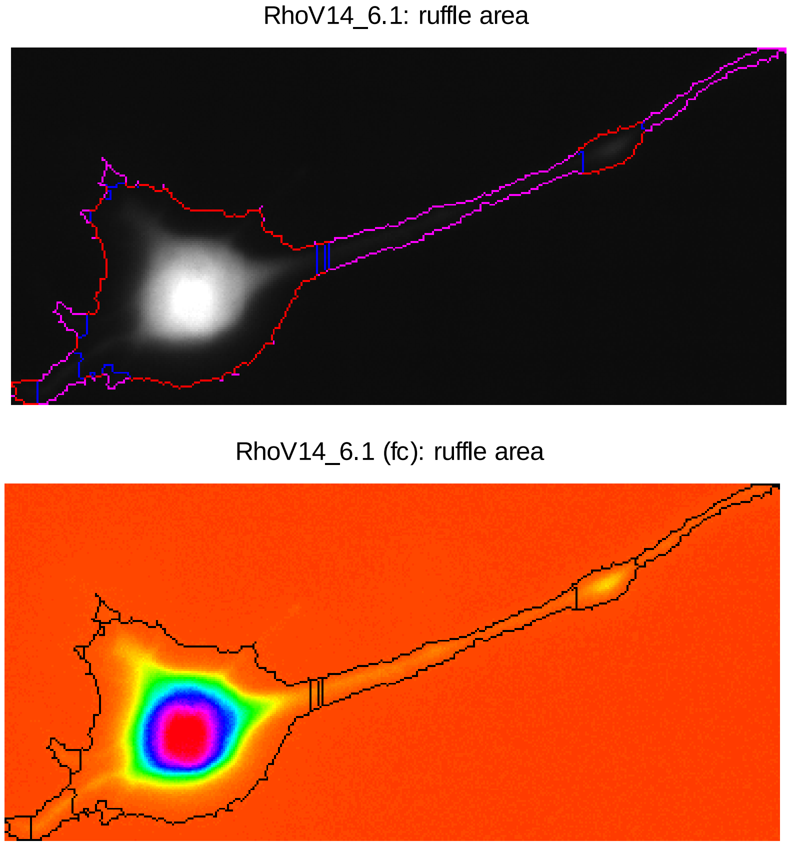
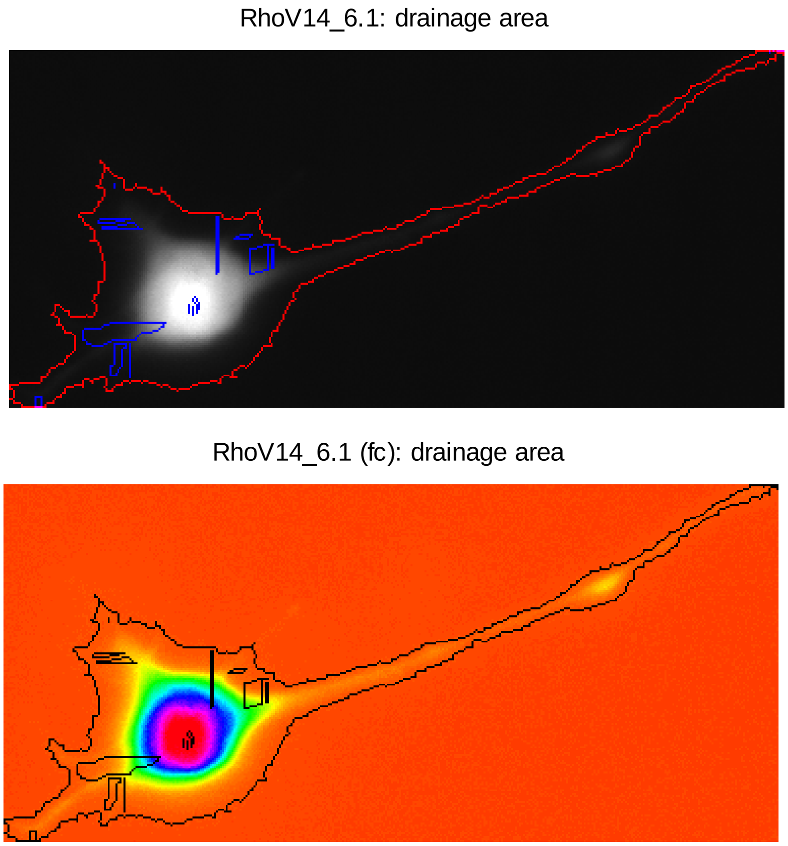


Figure S5: "Ruffle areas" as defined in (*1*) for cell RhoV14_6.1 in grayscale image (top) and false color image (bottom). Ruffle areas are areas of increased intensity near the cell segment border. They are shown in the top outlined in blue.

2.1 *Cell segment boundary:* This is the boundary defined by the user in CellSegmenter (see Figure S2). However, as described below, many features consider mathematically smoothed variants of this boundary (see item 3 below).

2.2 *GFP bright spot:* The bright spot comprises those pixels in a segment whose brightness is above the 90th percentile intensity of all pixels in the segment, with small, isolated areas of such pixels comprising 5% or less of the total bright spot area being eliminated. As with all feature elements, the bright spot is computed from the GFP image of a cell and therefore does not represent the cell nucleus, although nuclei may often occupy the bright spot (see Figure S2).

2.3 *Edges:* Edges demarcate sharp gradients of intensity. Edges are generally computed using the MatLab implementation of the 'canny' algorithm using default parameters. Features computed from edges (see below) provide information about the texture of the GFP intensity image within a segment (See Figure S3).



RhoV14_6.1: drainage area

RhoV14_6.1 (fc): drainage area

Figure S6: "Drainage areas" as defined in (*1*) for cell RhoV14_6.1 in grayscale image (top) and false color image (bottom). Drainage areas represent regions where the intensity gradient points inward, such that if intensity were represented in a third dimension as height, water would drain from their boundaries into the region. They are shown in the top outlined in blue.

2.4 Cell centers: Many features are computed with respect to cell 'centers'. There are several ways of defining these for a cell segment. Three that are used repeatedly and illustrated in Figure S2 are:

2.4.1 The *GFP centroid* is the geometric centroid of the binary image that represents the entire cell segment -- i.e., the point whose coordinates are the averages of all coordinates of pixels in the segment.

2.4.2 The *bright spot centroid* is the geometric centroid of the *GFP bright spot* (see item 2.2 above).

2.4.3 The *center of mass* of the segment is given by the mean x and y coordinates of pixels weighted by their intensity in the *segment mass image* (see **Image Normalization** above). It differs from the *GFP centroid* in that pixel intensities as well as positions are taken into account in determining the center of mass, whereas only positions are considered for the *GFP centroid*.

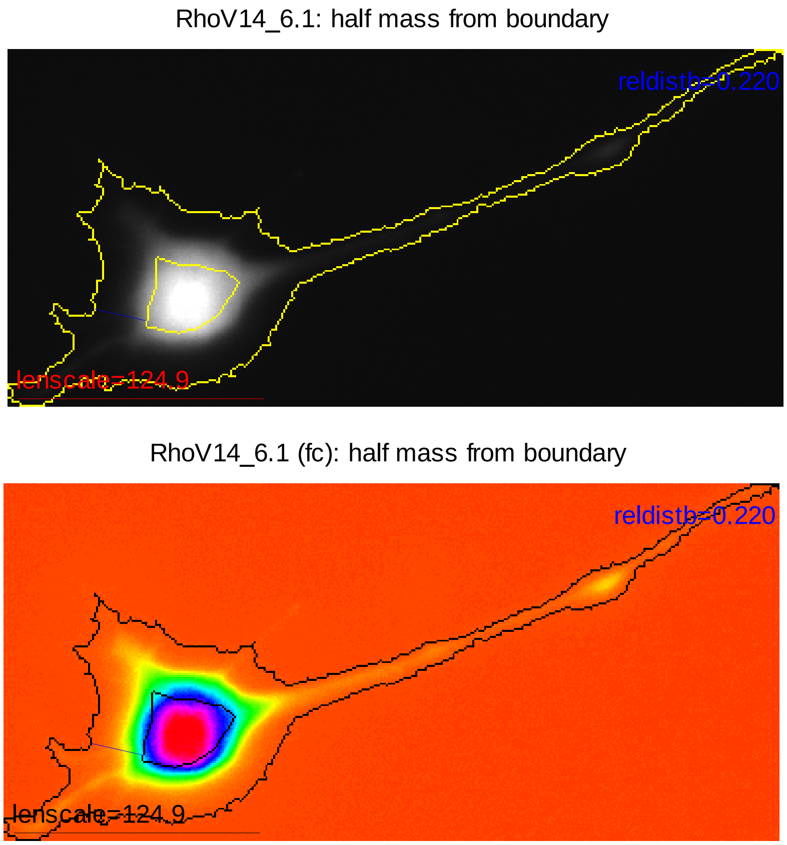

RhoV14_6.1: half mass from boundary
reldistb=0.220
lenscale=124.9
RhoV14_6.1 (fc): half mass from boundary
reldistb=0.220
lenscale=124.9

Figure S7: The *GFPHalfMassRelDistanceFromBoundary* feature illustrated for cell RhoV14_6.1 in grayscale (top) and false color (bottom). Pixels of distance ≤ *d* from the boundary are accumulated for increasing *d* until half of the intensity mass of the cell is captured. The value of the feature is *d* / *length_scale* (*length_scale* indicated in lower left). The blue line illustrates a representative distance *d* from the boundary. The region between the segment boundary and the interior border at distance *d* (interior yellow line, top panel) therefore contains 1/2 of the cell intensity mass.

2.5 *Reference units and relative feature values:* Many features are calculated as lengths or areas of geometric elements in a cell segment. In such cases, the feature as directly calculated has a scale that relates to the absolute size of the cell segment and its information content is conflated with cell segment size. However, by dividing the directly calculated feature by a standard cell segment-determined reference unit, a relative ratio is generated that reduces the dependency on cell size. For features directly calculated as areas, the reference unit is the entire area of the cell segment. For features directly calculated as lengths, the reference unit is called the *length_scale*. *length_scale* can be set to a number of possible length values in the MatLab program that calculates features, but for all work reported here and in the article text, *length_scale* has been set to the value of the MatLab EquivDiameter variable for the segment. EquivDiameter is defined as the diameter of the circle that has the same area as the cell segment.

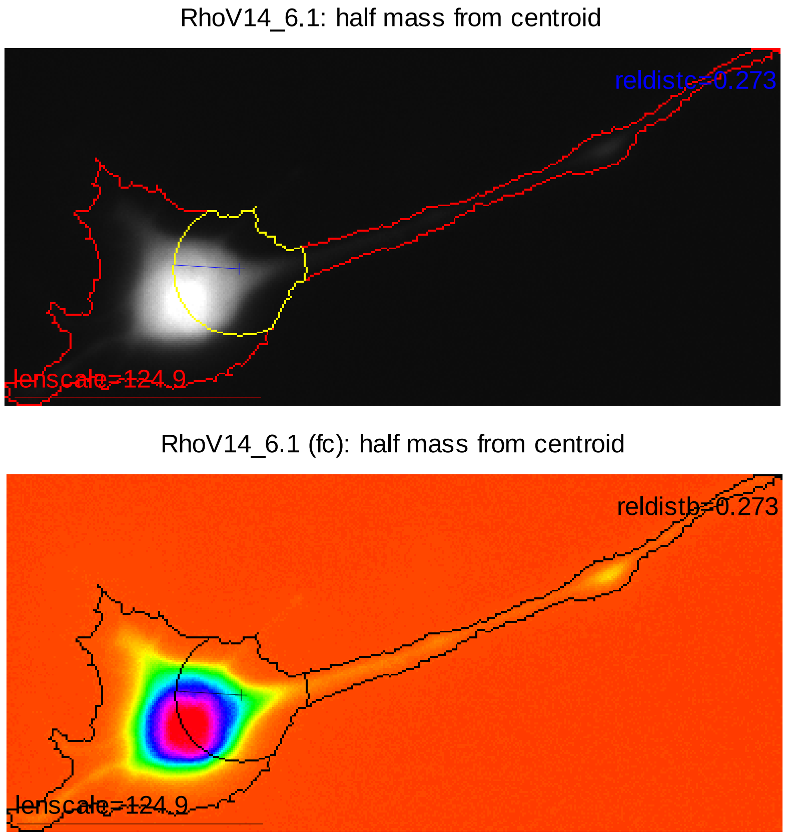

RhoV14_6.1: half mass from centroid
reldistb=0.273
lenscale=124.9
RhoV14_6.1 (fc): half mass from centroid
reldistb=0.273
lenscale=124.9

Figure S8: The *GFPHalfMassRelDistanceFromGFPCentroid* feature illustrated for cell RhoV14_6.1 in grayscale (top) and false color (bottom). Pixels of distance ≤ *d* from the *GFP centroid* are accumulated for increasing *d* until half of the intensity mass of the cell is captured. The value of the feature is *d* / *length_scale* (*length_scale* indicated in lower left). The blue line illustrates a representative distance *d* from the *GFP centroid*. The region within the interior partial circle around the *GFP centroid* (interior yellow line, top panel) therefore contains 1/2 of the cell intensity mass.

For example, after calculating edges within a segment (see item 2.3 above), mean edge length is an example of a directly calculable feature. As larger cell segments tend to have longer edges, the directly calculated absolute mean edge length will reflect the absolute size of the cell. However, dividing this value by *length_scale* will now give the mean edge length relative to the linear scale of the cell. This *relative* mean edge length will now have a reduced dependency on the absolute linear scale of the cell and be more descriptive of the texture of the GFP intensity image in the cell segment.

In several of the figures illustrating features (e.g., Figures S2 and S3), *length_scale* is indicated graphically as a horizontal line in the lower left corner along with a text description indicating the value of length_scale in units of pixels. In Figure S3, two examples of absolute vs. relative features are indicated: the total number of pixels and the mean edge length are absolute

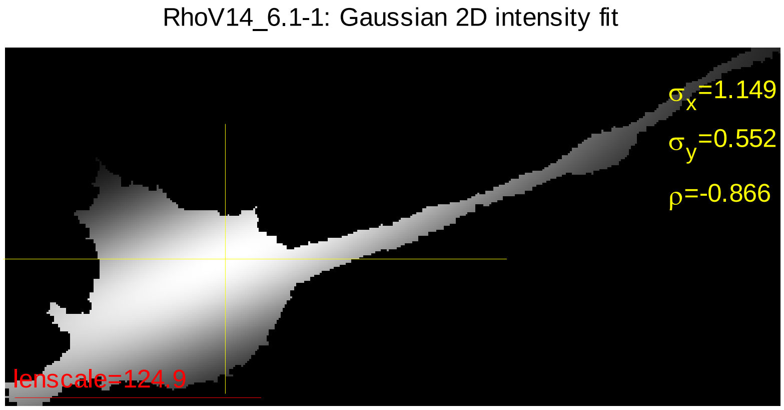


Figure S9: Gaussian 2D fit to pixel intensities of cell segment RhoV14_6.1. The intersection of the two yellow lines is the location of the mean of the best fitting Gaussian 2D surface to the 2D intensity profile of the cell segment. The horizontal yellow line extends a distance σ_x from the mean on either side, while the vertical yellow line similarly extends distance σ_y. The numerical values of σ_x and σ_y relative to *length_scale* (lower left) are indicated in the upper right, as is the ρ of the Gaussian.

features, while the total number of pixels / total cell segment area and the mean edge length / length_scale are relative features.

3 *Feature variations:* Above it was noted that there multiple centers may be defined in a cell, and also that length or area-based features can be reported as directly calculated 'absolute' values or as values relative to a standard reference length or area. Usually, in either of these cases, when there are multiple possibilities for calculating a feature, all of them are computed and reported. Thus, a large number of features are really slight variants of one another, differing in what centers they refer to and whether they are reported in absolute or relative terms. The reason for reporting multiple feature variants instead of just a single one is that one of the key purposes for computing features was to develop image classifiers. As it was not possible to know ahead of time which variant might be optimal for distinguishing between particular classes of images, we adopted the strategy of generating a large set of variants and letting the classifier training and construction logic determine the best set.

In addition to multiple centers and absolute vs. relative feature values, many features relating to cell shape are calculated from smoothed cell segment boundaries and a set of feature variants is computed and reported by varying the degree of smoothing applied to the cell segment boundary. Smoothing is accomplished by applying a Gaussian filter to the Fourier frequency spectrum of the boundary as per (*18*) (section 19.2.1.2, pp.490-1), so that high frequency 'noise' in the boundary eliminated. The σ of the Gaussian filter controls the degree of smoothing, whereby a large σ only yields a small degree of smoothing while a small σ yields a high degree. All features computed from smoothed boundaries are calculated with two degrees of smoothing, once with a large σ (LoSmooth) and again with a low σ (HiSmooth). These two degrees of smoothing provide related but different information about the shape of the boundary: The feature variants derived from the LoSmooth boundary

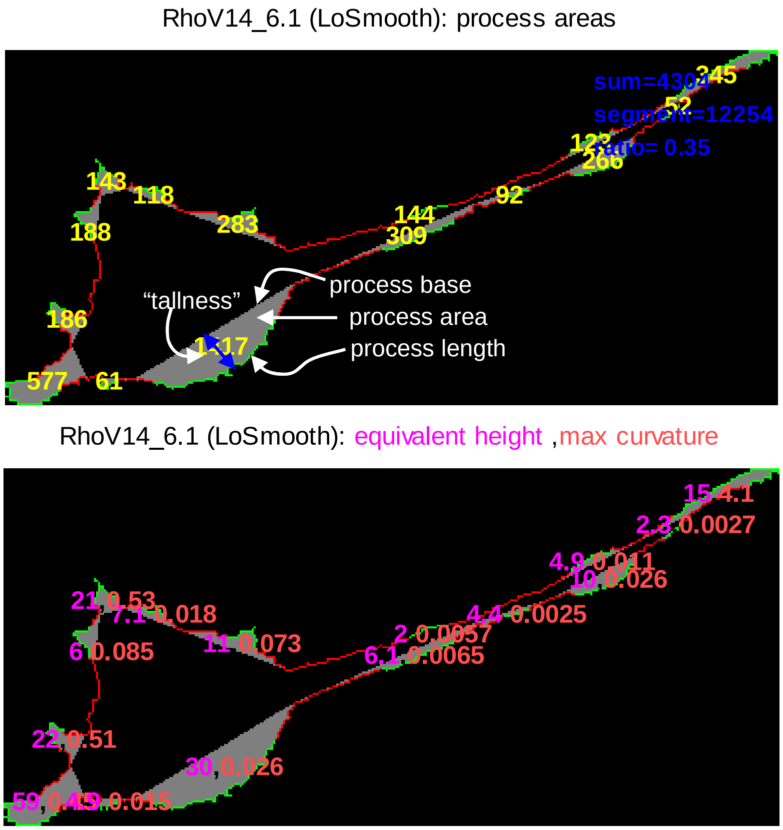


Figure S10: Process analysis for cell segment RhoV14_6.1 for LoSmooth boundary (see text). Here processes are drawn on the original boundary of the cell, not the 'LoSmooth'ed boundaries (see Figure S4). Green arcs on the boundary represent arcs of positive curvature on the 'LoSmooth'ed boundary, and red arcs represent arcs of negative curvature; each process extends from a green arc to the points of minimum (i.e., most negative) negative curvature on the abutting red arcs. Several features associated with processes are indicated in the top panel: Each process has a length, base, an area, and an "equivalent height" or "tallness" (illustrated, conceptually by the blue double arrow). Values in the top panel give the area in pixels of each process. Values in the bottom panel give the equivalent height and maximum positive curvature ("sharpness") of each process. Printed in blue (top panel) are the sum of all process areas, the total cell segment area, and the ratio of the two (the feature *LoSmoothBndUndulationTotalRelativeArea*).

provide information relating to the local shape of the cell boundary -- e.g., small protrusions of or undulations in the boundary -- while the variants derived from the HiSmooth boundary describe only large-scale undulations of the boundary and therefore describe overall cell shape. Specifically, LoSmooth boundaries smooth with σ = P/70, while the HiSmooth boundaries smooth with σ = P/200, where P = the number of pixels in the cell segment perimeter. In effect, σ = P/70 effectively eliminates frequencies that are >= 2*σ (frequency units are 1/P = 1 cycle per P pixels), meaning that frequencies of 1/35 or higher (1 cycle in 35 pixels) within the boundary are eliminated, while σ = P/200 similarly leads to the elimination of

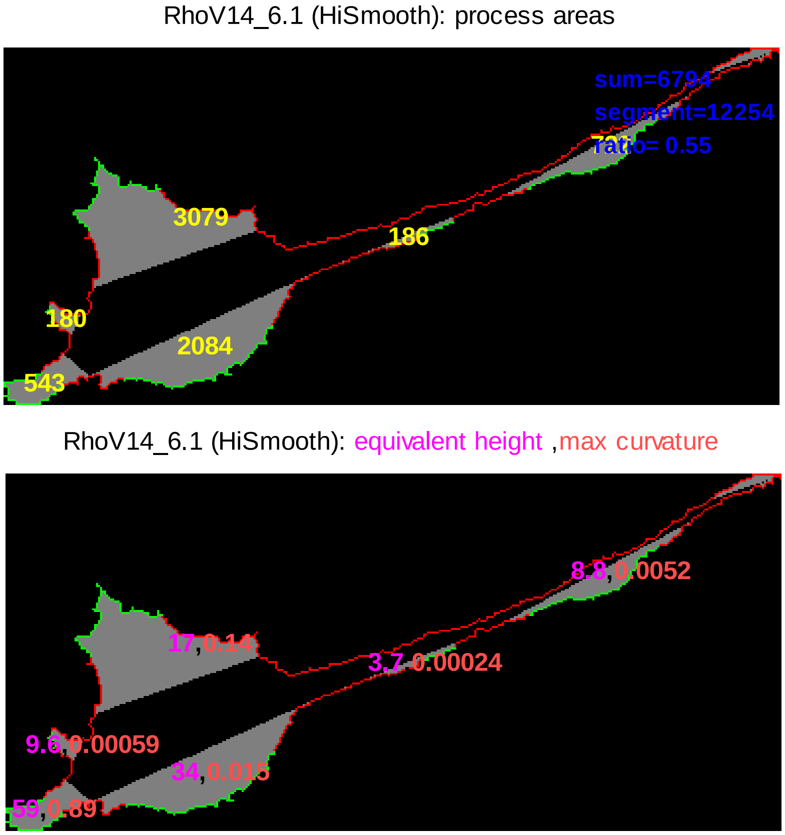


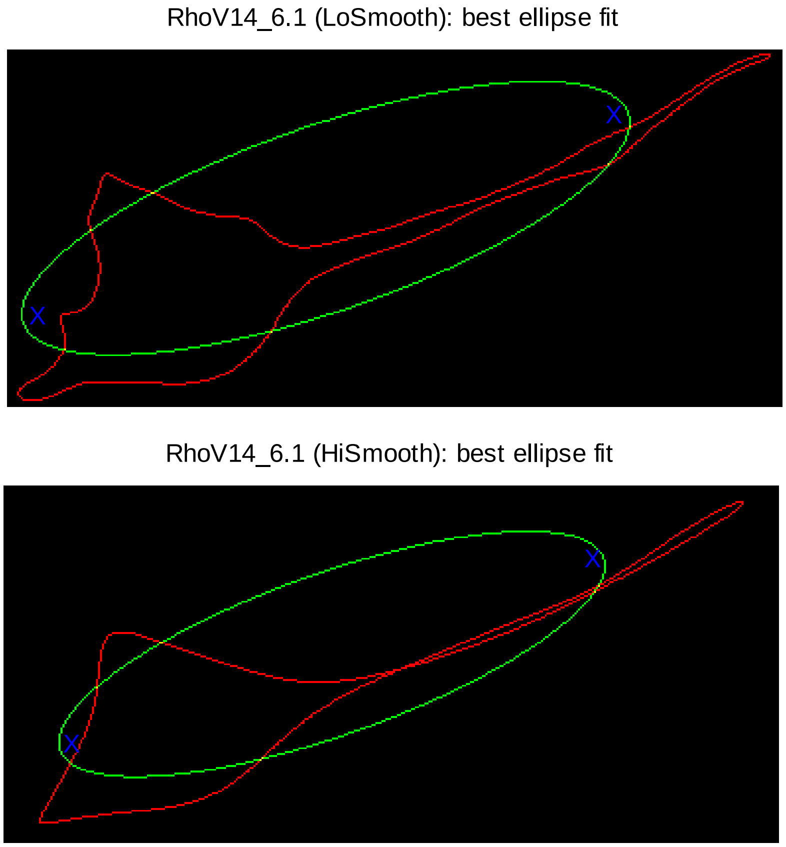
Figure S11: Process analysis for cell segment RhoV14_6.1 for HiSmooth boundary (see text). The panels and information content of this figure are the same as in Figure S10 except for their reference to HiSmooth vs. LoSmooth boundaries.

frequencies of 1 cycle in 100 pixels or higher. These values were set after early experimentation with a small number of images. Figure S4 illustrates LoSmooth and HiSmooth smoothing.

Feature Classes

As the 154 features have multifaceted relationships to each other and to the feature elements from which they are computed, classifying features into categories is difficult. Nevertheless, for purposes of discussion, a rough-and-ready classification is presented here. Each class is given a code in parentheses that is used to describe individual features (below).

I *Status/quality indicators (STATUS):* This class describes the nine status and quality indicators that were mentioned above. These features were not used in defining classifiers and are removed from normalized versions of the feature data.



RhoV14_6.1 (LoSmooth): best ellipse fit

RhoV14_6.1 (HiSmooth): best ellipse fit

Figure S12: Best ellipse fits to LoSmooth (top) and HiSmooth (bottom) boundaries for cell segment RhoV14_6.1, showing foci of the ellipses (blue Xs).

II *Basic morphology (BASIC):* These features describe the basic dimensions and geometry of the cell segment. Examples include Area, MajorAxisLength, and EquivDiameter (which, as noted in 2.5 above, is the reference *length_scale* used for computing relative linear features). GFP image intensities are not considered in any of these features. Many of these features are computed from built-in MatLab image analysis functions.

III *Cell center offsets (CENTER):* The three cell centers described above in 2.4 are all based on different elements of an image. Unlike the *GFP centroid*, both the *GFP bright spot centroid* and *center of mass* take into account GFP intensity information, but do so in different ways (locations of the brightest pixels vs. overall intensity distribution). The relative offsets of these various centers to each other therefore provide information about asymmetries in the distribution and location of bright pixels relative to cell geometry. See Figure S2 for illustrations of the three cell centers.

IV *GFP intensity distribution features (INTENSITY):* This class comprises straightforward features such as mean and standard deviation of GFP intensity, but also several features that are more specific to cell morphology. Included among these are

IV.1 features related to fragmentation of the *GFP bright spot*

IV.2 reconstructions of previously published features that describe cell morphology, in particular statistics for "ruffle areas", "internal drainage", "moment of inertia," and "multivariate kurtosis" from (*1*). These were described as informative features relating to Rac1 phenotypes in CHO cells: Ruffle areas are small hills of intensity near the borders of cells, drainage areas are valleys of intensity internal to cells, while moment of inertia and multivariate kurtosis describe the overall shape of the spatial GFP distribution. Ruffle areas are illustrated in Figure S5, and drainage areas are illustrated in Figure S6. The CHO cells analyzed in (*1*) were generally regular in shape compared to the cells analyzed in our study, so that these features may operate differently here. For instance, many cells in our study exhibit long narrow processes and ruffle areas tend to get caught up in these processes, a situation which does not arise in the more regular CHO cells (see Figure S5).

IV.3 several other experimental statistics that provide information on other aspects of the intensity distribution:

IV.3.1 *Half-mass* statistics describe how close to a cell center or the segment boundary one has to get to capture 50% of the total intensity of the cell. These features therefore measure overall concentration of intensity near the boundary or a cell center. (See Figures S7 and S8).

IV.3.2 *Edge statistics* provide information about intensity edges and textures within the cell segment. (See Figure S3.)

IV.3.3 *Gaussian 2D Fit statistics* are based on a fit of a 2D Gaussian distribution to the GFP intensity distribution. These features provide general information about the shape and degree of fall-off in the intensity distribution. Mathematically, the fit of intensity distributions to Gaussian 2D distributions is hard to perform for irregular cell shapes, and a large penalty is used to constrain the mean of the fit Gaussian to the *GFP bright spot*. However, for *very* irregular cell shapes, even the large penalty term may fail to ensure a good fit.

IV.3.4 *Mutual information statistics* report on the amount of mutual information that exists between GFP intensity and locations within the cell, and is intended as an easy-to-calculate measure of the 'texture' of the GFP intensity distribution.

V *Boundary analysis (BOUNDARY):* This is a large class of features that relate to the analysis of the cell boundary. As noted above (item 3 in **General Aspects of Features**), two variants of most of these features are computed, one with a high degree and the other with a low degree of smoothing. Broadly speaking, most of the *BOUNDARY* features analyze the sizes, degrees of sharpness, and numbers of regions of positive and negative curvature in smoothed cell boundaries. These can be interpreted as describing the number, size, and sharpness of *processes* or *undulations* of the cell boundary. (Terminologically, a process can be thought of as a particularly sharp undulation, but there is no intrinsically meaningful numerical threshold on size or sharpness for distinguishing them. In practice, we tend to use the term 'process' for any subset of undulations defined by a chosen threshold.) Processes and undulations are identified with entire contiguous arcs of positive curvature, that are then extended into the abutting arcs of negative curvature on either side up until the points of largest (i.e., most negative) negative curvature. Several elements and parameters (below; see also Figures S10 and S11) are considered in the calculation of process-oriented BOUNDARY features, including:

V.1 *Process (undulation) base:* the line segment joining the endpoints of the process arc. The length of this line is the computed feature.

V.2 *Process (undulation) area:* the area of the region bounded by the process arc and the process (undulation) base.

V.3 *Process (undulation) length:* the length of the process arc.

V.4 *Sharpness:* the maximum positive curvature on the process (undulation) arc

V.5 *Equivalent height or "tallness":* Twice the area of the process divided by the length of the process base. "Tallness" is thus computed as if the process were approximated by a triangle constructed on the process base.

Several of these features are illustrated in Figures S10 and S11.

VI Ellipticity features (ELLIPTICITY): These features are computed from the ellipse that best fits a smoothed boundary. The reason for fitting ellipses to *smoothed* boundaries is to avoid distracting the ellipse fitting process by minor details of segment boundaries that are smoothed away. (Several BASIC features such as MajorAxisLength, MinorAxisLength, and Eccentricity derive from MatLab fitting of ellipses to unsmoothed boundaries, so this information is *also* available.) MatLab ellipse calculation features are supplemented by custom code to compute additional ellipse-related information including the location of the foci of the ellipse and the error of the best fit, which are used to provide additional features. (See Figure S12.)

Breakdown of Features into Classes

The numbers of the 154 features in each class is given in the following table.

Feature class	**Number**
STATUS	9
BASIC	6
CENTER	9
INTENSITY	36
ELLIPTICITY	8
BOUNDARY	86
Total	154

Definitions of Individual Features

Recall that the length scale (*length_scale*) used for relative distance measurements is EquivDiameter. The notation (L) indicates a feature adapted from (*1*).

STATUS features

These status / quality indicators are produced in the course of feature analysis in order to gauge the quality or success of various aspects of feature processing. Although they qualify the meaning of related calculated features, and provide some information about the cell segment, they are not used in subsequent feature analysis or in the construction and training of cell segment classifiers.

SegmentationThreshold: The threshold used in CellSegmenter to define the cell segment being analyzed.

FractionBorderPixel: The fraction of the number of the pixels in the cell segment perimeter that are on the image border.

FractionBarrierPixel: The fraction of the number of pixels in the cell segment perimeter that are on barriers drawn in the image by the CellSegmenter user.

RuffleAnalysisStatus (L): A status code that describes whether ruffle analysis was successful.

DrainageStatus (L): A status code that describes whether drainage analysis was successful.

GFPGauss2DFitStatus: A status code that describes whether the Gaussian 2D fit to the segment intensity profile was successful.

LoSmoothEllipticityStatus, HiSmoothEllipticityStatus: A status code that describes whether the best fit of an ellipse to the cell segment smoothed boundary was successful.

SegmentProcessingTime: The amount of time it took in seconds to perform feature analysis for the cell segment.

BASIC features

Area: The area of (total number of pixels in) the cell segment.

Solidity: The ratio of the area of the cell segment to the area of the convex hull of the cell segment. Solidity ranges between 0 and 1. It is 1 for a perfectly convex segment and smaller than 1 for segments that have regions of concavity.

Eccentricity: The eccentricity of the ellipse that best fits the cell segment boundary (as calculated by MatLab built-in functions on the unsmoothed cell segment boundary).

MajorAxisLength: The length of the major axis of the ellipse that best fits the cell segment boundary (as calculated by MatLab built-in functions on the unsmoothed cell segment boundary).

MinorAxisLength: The length of the minor axis of the ellipse that best fits the cell segment boundary (as calculated by MatLab built-in functions on the unsmoothed cell segment boundary).

EquivDiameter: The length of the diameter of the perfect circle that had the same area as the cell segment. As noted above, this value is used as the *length_scale* of the cell segment that is used for reporting relative distances.

CENTER features

See section 2.4 in **General Aspects of Features** for details on center calculations. See Figure S2 for illustrations of cell segment centers.

GFPBrightSpotGFPCentroidRelOffset: The distance between the *GFP bright spot centroid* and *GFP centroid*, relative to *length_scale*.

GFPCentroidGFPCenterOfMassRelOffset: The distance between the *GFP centroid* and the cell segment *center of mass*, relative to *length_scale*.

GFPBrightSpotGFPCenterOfMassRelOffset: The distance between the *GFP bright spot centroid* and cell segment *center of mass*, relative to *length_scale*.

LoSmoothGFPCentroidClosestFocusRelOffset, HiSmoothGFPCentroidClosestFocusRelOffset: The distance between the *GFP centroid* and the closest focus of the best-fit ellipse generated for *ELLIPTICITY* features, relative to *length_scale*.

LoSmoothGFPCenterOfMassClosestFocusRelOffset, HiSmoothGFPCenterOfMassClosestFocusRelOffset: The distance between the cell segment *center of mass* and the closest focus of the best-fit ellipse generated for *ELLIPTICITY* features, relative to *length_scale*.

LoSmoothGFPBrightSpotClosestFocusRelOffset, HiSmoothGFPBrightSpotClosestFocusRelOffset: The distance between the *GFP bright spot centroid* and the closest focus of the best-fit ellipse generated for *ELLIPTICITY* features, relative to *length_scale*.

INTENSITY features

Figures S2, S3, S5, S6, S7, and S8 illustrate the intensity profile of a cell segment along with several features computed from the intensity profile and distribution.

MeanIntensity: The mean of the pixel intensities for all pixels within the cell segment. (Pixel intensities are normalized to a maximum of 1 over the entire image containing the segment.)

StdIntensity: The standard deviation of the pixel intensities for all pixels within the cell segment.

90thPercentileIntensity: The 90th percentile intensity of all pixels within the cell segment. Note that this is the intensity threshold used to define the cell segment's *GFP bright spot*.

GFPBrightSpotMajorSegments: The number of distinct regions within the cell segment that consist of pixels exceeding the *GFP bright spot* intensity threshold. As described above, small bright spot segments are eliminated when defining the bright spot, and this feature counts only those bright spot segments that survive this clean-up process (hence the phrase "MajorSegment" within the feature name). It is possible (but unlikely) that no segments could remain after this clean-up.

GFPBrightSpotTotalArea: The total area of (total number of pixels in) the *GFP bright spot*, no matter how many segments it may contain. Since the bright spot is defined by the 90^{th} percentile intensity threshold, this value should be close to 10% of the total segment area. However, the clean-up of small bright spot segments and the discreteness of the number of pixels in the cell may lead to deviations from this value.

GFPBrightSpotMajorSegmentAreaMean: The mean of the areas of the 'major segments' of the *GFP bright spot* as described in *GFPBrightSpotMajorSegments* above, after clean-up of small bright spot areas.

GFPBrightSpotMajorSegmentAreaCV: The coefficient of variation of the areas of the 'major segments' of the *GFP bright spot*, after clean-up of small bright spot areas. This feature is intended to provide information on the degree of variation in size of dispersed bright spot major areas.

GFPBrightSpotMajorSegmentMaxMinSeparation: If there are multiple *GFP bright spot* major segments, each has a minimum distance to the others (i.e., the shortest distance between any pixel in one segment to the pixels in all other segments). This feature reports the maximum of these minimum distances relative to *length_scale*, and therefore provides information about the degree of dispersion of bright spot major segments when there is more than one.

GFPCenterOfMassGFPMomentOfInertia (L): The moment of inertia of the GFP intensity distribution computed with reference to the *GFP center of mass*. The formula for moment of inertia is $\Sigma m_i \cdot r_i^2$ where i ranges over all pixels in the cell segment, m_i is the 'mass' at the pixel (see section 2.4 in **General Aspects of Features** for details on *center of mass* calculations), and r_i is the Euclidean distance between the pixel and the *center of mass*. This feature was not normalized to *length_scale* because there is no indication of any such normalization in (*1*).

GFPCentroidGFPMomentOfInertia (L): The moment of inertia of the GFP intensity distribution computed with reference to the *GFP centroid*. See *GFPCenterOfMassGFPMomentOfInertia* for other details on this calculation.

GFPBrightSpotGFPMomentOfInertia (L): The moment of inertia of the GFP intensity distribution computed with reference to the *GFP bright spot centroid*. See *GFPCenterOfMassGFPMomentOfInertia* for other details on this calculation.

GFPMultivariateKurtosis (L): The multivariate kurtosis of the intensity distribution, computed according to formula 3.5 of (*19*) (cited by (*1*)), which calculates it as $\beta_{2p} = \mathbf{E}\left(((X-\mu)'\boldsymbol{\Sigma}^{-1}(X-\mu))^2 \right)$. Here X is the random variable that describes locations (x,y) of cell mass, μ is the mean E(X) of X (equal to the *center of mass*), and Σ is the covariance matrix of X. See the section **Image Normalization** for details on the *segment mass image* that is used to compute this value.

GFPHalfMassRelDistanceFromBoundary: See section IV.3 of **Feature Classes** above for general information on half-mass features. This feature is illustrated in Figure S7.

GFPHalfMassRelDistanceFromGFPCentroid: See section IV.3 of **Feature Classes** above for general information on half-mass features. This feature is illustrated in Figure S8.

GFPHalfMassRelDistanceFromGFPCenterOfMass: A variant of *GFPHalfMassRelDistanceFromGFPCentroid* that uses the *GFP center of mass* instead of the *GFP centroid* as the center of for computing half-mass distance. See section IV.3 of **Feature Classes** above for general information on half-mass features. The *GFP centroid* version of the feature is illustrated in Figure S8.

GFPHalfMassRelDistanceFromGFPBrightSpotCentroid: A variant of *GFPHalfMassRelDistanceFromGFPCentroid* that uses the *GFP bright spot centroid* instead of the *GFP centroid* as the center of for computing half-mass distance. See section IV.3 of **Feature Classes** above for general information on half-mass features. The *GFP centroid* version of the feature is illustrated in Figure S8.

RuffleArea (L): The total number of pixels in "ruffle areas", which are small 'hills' of intensity that are close to the cell boundary.

RufflePixSum (L): The total of the intensity of pixels in the "ruffle areas."

RuffleVolume (L): Viewing intensity values as defining a surface above the 2D planar region of the cell segment, and focusing on a single ruffle area, the amount of volume is calculated between the ruffle area surface and the horizontal plane that is set at the highest intensity on the border between the ruffle area and cell interior. The feature is the sum of these volumes over all ruffle areas.

DrainageArea (L): The total number of pixels in "internal drainage areas," which are small 'depressions' of intensity within the cell segment interior.

DrainagePixSum (L): The total of the intensity of pixels in the "drainage areas."

GFPEdgeNumber: The number of edges inside of the cell segment. See section 2.3 of **General Aspects of Features** for information on computation of edges, and Figure S3 for an illustration of edges and edge feature calculations.

GFPEdgeTotalPixels: The total number of pixels in all edges inside of the cell segment. See section 2.3 of **General Aspects of Features** for information on computation of edges, and Figure S3 for an illustration of edges and edge feature calculations.

GFPEdgePixelDensity: The total number of pixels in all edges inside of the cell segment, divided by the cell segment area (i.e., the total number of pixels in the cell segment). See section 2.3 of **General**

Aspects of Features for information on computation of edges, and Figure S3 for an illustration of edges and edge feature calculations.

GFPEdgeMeanLength: The mean number of pixels in the edges within the cell segment. See section 2.3 of **General Aspects of Features** for information on computation of edges, and Figure S3 for an illustration of edges and edge feature calculations.

GFPEdgeMeanRelativeLength: The mean number of pixels in the edges within the cell segment, divided by *length_scale*. See section 2.3 of **General Aspects of Features** for information on computation of edges, and Figure S3 for an illustration of edges and edge feature calculations.

GFPIntensityLocationMutualInformation_5_15_15, GFPIntensityLocationMutualInformation_8_15_24, GFPIntensityLocationMutualInformation_5_20_15, GFPIntensityLocationMutualInformation_8_20_24: Mutual information between locations and intensity is computed by overlaying a grid on the cell with a specific grid element length (ElementSize). Grid elements that contain fewer than a certain number of segment pixels are excluded (MinPixels). Probability distributions for the intensity distribution over the entire cell segment, and for each grid element under consideration, are constructed from histograms based on a range of frequency bins (IntensityMeshSize). Using this scheme, mutual information between locations and intensities is computed by the standard formula I(I;L)=H(X) - H(X|L), where X = intensity distribution, L = location distribution as given by intensities (cell 'mass', as above) where H = entropy. The feature named *GFPIntensityLocationMutualInformation_X_Y_Z* is calculated with an IntensityMeshSize = X, ElementSize = Y, and MinPixels = Z.

GFPGauss2DFitMeanResidual: See section IV.3 of **Feature Classes** for information on GFP Gaussian 2D fit features, and Figure S9 for an illustration. This feature is the total residual of the 2D Gaussian fit divided by the number of pixels in the cell segment, a measure of how well the cell intensity profile is represented by a Gaussian. Note that the fit of intensities to the Gaussian is performed in log space. Low values indicate a good fit, while higher values indicate fits that are not as good.

GFPGauss2DFitCorrelation: See section IV.3 of **Feature Classes** for information on GFP Gaussian 2D fit features, and Figure S9 for an illustration. This feature is reported as the Gaussian ρ parameter (i.e., correlation) of the best-fit 2D Gaussian to the cell's intensity profile.

GFPGauss2DFitRelativeSigmaRow: See section IV.3 of **Feature Classes** for information on GFP Gaussian 2D fit features, and Figure S9 for an illustration. *Values reported for this feature are in error!!* This feature is supposed to present the σ_y parameter (standard deviation in the row (y) direction) of the best-fit 2D Gaussian to the cell's intensity profile, relative to *length_scale*. However due to a coding error, the values accumulated for this feature for all cell segments analyzed in this study are actually the μ_y parameters (means in the row (y) direction), rather than the σ_y, relative to *length_scale*.

GFPGauss2DFitRelativeSigmaCol: See section IV.3 of **Feature Classes** for information on GFP Gaussian 2D fit features, and Figure S9 for an illustration. This feature presents the σ_x parameter (standard deviation in the column (x) direction) of the best-fit 2D Gaussian to the cell's intensity profile, relative to *length_scale*.

GFPGauss2DFitRelativeOffsetMeanFromSegCentroid: The distance between the location of the mean of the best-fit 2D Gaussian and the *GFP centroid* of the cell, relative to *length_scale*.

GFPGauss2DFitRelativeOffsetMeanFromBrightSpotCentroid: The distance between the location of the mean of the best-fit 2D Gaussian and the *GFP bright spot centroid* of the cell, relative to *length_scale*.

ELLIPTICITY features

See section IV of **Feature Classes** for general information on ELLIPTICITY features. Best-fit ellipses to smoothed boundaries are illustrated in Figure S12.

LoSmoothEccentricity, HiSmoothEccentricity: The eccentricity of the ellipse that best fits the smoothed boundary, as computed by built-in MatLab functions.

LoSmoothMajorAxisLength, HiSmoothMajorAxisLength: The major axis length of the ellipse that best fits the smoothed boundary, as computed by built-in MatLab functions.

LoSmoothMinorAxisLength, HiSmoothMinorAxisLength: The minor axis length of the ellipse that best fits the smoothed boundary, as computed by built-in MatLab functions.

LoSmoothEllipticity, HiSmoothEllipticity: The residual of the best-fit ellipse divided by the number of pixels in the smoothed cell boundary that was fit. Computation of these features requires additional calculations beyond MatLab built-in functions, which do not return ellipse foci or fit residuals; however the MatLab-based values of eccentricity, major axis length, minor axis length, and orientation, are used as input to this process. (A variant algorithm which computed these values *ab initio* worked well but sometimes gave results that differed greatly from MatLab-based calculations. While these seemed to be genuinely ambiguous cases, the *ab initio* calculations were avoided so as to preserve general consistency with MatLab-based calculations.) This feature gives a measure of the degree to which the smoothed boundary is elliptical. Low values indicate a good fit, while higher values indicate fits that are not as good.

BOUNDARY features

See section V of **Feature Classes** and section 3 of **General Aspects of Features** for general information on BOUNDARY features. Figure S10 and S11 present illustrations of BOUNDARY process analysis.

Many BOUNDARY features derive from computed curvatures of points on the boundary. Curvatures (κ) are computed using the standard formula $\kappa = \dfrac{\dot{x}\cdot\ddot{y} - \dot{y}\cdot\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}$, where first and second derivatives of *x* and *y* are themselves computed from the smoothed, parameterized boundary.

LoSmoothBndNormIntegratedAbsAngle, HiSmoothBndNormIntegratedAbsAngle: The integral of the absolute value of the increment of tangent angle dθ over the parameterized smoothed cell boundary, taken over the entire boundary, divided by the same value computed for a perfect circle of the same perimeter as the cell boundary. dθ is computed during curvature calculations using the formula $d\theta = \dfrac{\dot{x}\cdot\ddot{y} - \dot{y}\cdot\ddot{x}}{\dot{x}^2 + \dot{y}^2}$. In theory, dθ is constant along a perfect circle, while, for irregular boundaries, dθ alternates between regions of positive and then negative value as one moves alternately through boundary regions with positive and negative curvature; however, (again, in theory), the integral of dθ over a cell boundary (∫dθ) should always equal 2π no matter how irregular the boundary because the positive and negative contributions to ∫dθ accumulated over these regions should cancel out, with the final value of ∫dθ ultimately representing only a net sweep of the tangent through 360° over the entire boundary. By contrast, positive and negative values do not cancel out in this way when the *absolute value* of dθ is integrated (∫|dθ|), and ∫|dθ| should increase to larger and larger values with increasingly convoluted boundaries. In practice, calculations of both ∫dθ and ∫|dθ| are heavily affected by the discreteness of images, the effectiveness and degree of smoothing, and the size of the boundary. To partially correct these artifacts, the ∫|dθ| computed for a smoothed cell boundary is normalized by dividing it by the value of ∫|dθ| computed for a perfect circle with the same perimeter as the cell boundary that has been smoothed in the same way as the cell boundary, and this ratio is reported as the feature. A value near 1 indicates a simple cell boundary, while values > 1 indicate boundaries that are increasingly complicated. We thank Steve Altschuler for discussions regarding this feature.

LoSmoothBndUndulationCount, HiSmoothBndUndulationCount: The total number of undulations (regions of positive curvature) in the smoothed cell boundary.

LoSmoothBndUndulationTotalRelativeArea, HiSmoothBndUndulationTotalRelativeArea: The total area contained in all undulations of the smoothed cell boundary, divided by the total area of the cell segment. See Figures S10 and S11 for illustrations.

LoSmoothBndProcessesGE0.5, HiSmoothBndProcessesGE0.5: The number of undulations having a maximum positive curvature of at least 0.5.

LoSmoothBndProcessesGE1, HiSmoothBndProcessesGE1: The number of undulations having a maximum positive curvature of at least 1.

LoSmoothBndCurvatureSharpestProcess, HiSmoothBndCurvatureSharpestProcess: The maximum positive curvature found on the smoothed cell boundary, indicating the sharpest process where sharpness is defined by curvature.

LoSmoothAreaSharpestProcess, HiSmoothAreaSharpestProcess: The area of the process with the maximum positive curvature found on the smoothed cell boundary.

LoSmoothRelativeAreaSharpestProcess, HiSmoothRelativeAreaSharpestProcess: The area of the process with the maximum positive curvature found on the smoothed cell boundary, divided by the total area of the cell segment.

LoSmoothBndCurvature2ndSharpestProcess, HiSmoothBndCurvature2ndSharpestProcess: The maximum positive curvature of the process with the second highest positive curvature found on the smoothed cell boundary, indicating the second sharpest process where sharpness is defined by curvature alone. Note that all boundaries will have at least one region of positive curvature, and therefore a sharpest process, but not all boundaries will have two. A value of 0 for this feature indicates that there is no second sharpest process.

LoSmoothArea2ndSharpestProcess, HiSmoothArea2ndSharpestProcess: The area of the second sharpest process found on the smoothed cell boundary, where sharpness is defined by curvature. A value of 0 for this feature indicates that there is no second sharpest process.

LoSmoothRelativeArea2ndSharpestProcess, HiSmoothRelativeArea2ndSharpestProcess: The area of the second sharpest process found on the smoothed cell boundary, divided by the total area of the cell segment, where sharpness is defined by curvature. A value of 0 for this feature indicates that there is no second sharpest process.

LoSmoothBndAngleSharpestProcessesGFPCentroid, HiSmoothBndAngleSharpestProcessesGFPCentroid: Where there is both a sharpest and a second sharpest process, the angle (in degrees) subtended by the points on the unsmoothed boundaries corresponding to the maximum curvature points on the smoothed boundary processes, relative to the cell segment *GFP centroid.* Bipolar cells have a value of this feature that is close to 180. A value of 0 for this feature indicates that there is no second sharpest process.

LoSmoothBndAngleSharpestProcessesGFPCenterOfMass, HiSmoothBndAngleSharpestProcessesGFPCenterOfMass: A feature variant of *LoSmoothBndAngleSharpestProcessesGFPCentroid* (and its HiSmooth variant) where the cell center used for measuring the subtended angle is the *GFP center of mass*.

LoSmoothBndAngleSharpestProcessesGFPBrightSpotCentroid, HiSmoothBndAngleSharpestProcessesGFPBrightSpotCentroid: A feature variant of *LoSmoothBndAngleSharpestProcessesGFPCentroid* (and its HiSmooth variant) where the cell center used for measuring the subtended angle is the *GFP bright spot centroid*.

LoSmoothHeightTallestProcess, HiSmoothHeightTallestProcess: The 'equivalent height' or 'tallness' of the tallest undulation on the smoothed cell boundary.

LoSmoothRelativeHeightTallestProcess, HiSmoothRelativeHeightTallestProcess: The 'equivalent height' or 'tallness' of the tallest undulation on the smoothed cell boundary, relative to *length_scale*

LoSmoothAreaTallestProcess, HiSmoothAreaTallestProcess: The area of the tallest undulation on the smoothed cell boundary.

LoSmoothRelativeAreaTallestProcess, HiSmoothRelativeAreaTallestProcess: The area of the tallest undulation on the smoothed cell boundary, relative to total cell segment area.

LoSmoothBaseTallestProcess, HiSmoothBaseTallestProcess: The base of the tallest undulation on the smoothed cell boundary.

LoSmoothRelativeBaseTallestProcess, HiSmoothRelativeBaseTallestProcess: The base of the tallest undulation on the smoothed cell boundary, relative to *length_scale*.

LoSmoothHeight2ndTallestProcess, HiSmoothHeight2ndTallestProcess: The 'equivalent height' or 'tallness' of the *second* tallest undulation on the smoothed cell boundary. Note that if there is only one arc of positive curvature on the boundary, there is no second tallest undulation and the value of this feature is 0.

LoSmoothRelativeHeight2ndTallestProcess, HiSmoothRelativeHeight2ndTallestProcess: The 'equivalent height' or 'tallness' of the *second* tallest undulation on the smoothed cell boundary, relative to *length_scale*. Note that if there is only one arc of positive curvature on the boundary, there is no second tallest undulation and the value of this feature is 0.

LoSmoothArea2ndTallestProcess, HiSmoothArea2ndTallestProcess: The area of the *second* tallest undulation on the smoothed cell boundary. Note that if there is only one arc of positive curvature on the boundary, there is no second tallest undulation and the value of this feature is 0.

LoSmoothRelativeArea2ndTallestProcess, HiSmoothRelativeArea2ndTallestProcess: The area of the *second* tallest undulation on the smoothed cell boundary, relative to total cell segment area. Note that if there is only one arc of positive curvature on the boundary, there is no second tallest undulation and the value of this feature is 0.

LoSmoothBase2ndTallestProcess, HiSmoothBase2ndTallestProcess: The base of the *second* tallest undulation on the smoothed cell boundary. Note that if there is only one arc of positive curvature on the boundary, there is no second tallest undulation and the value of this feature is 0.

LoSmoothRelativeBase2ndTallestProcess, HiSmoothRelativeBase2ndTallestProcess: The base of the *second* tallest undulation on the smoothed cell boundary, relative to *length_scale*. Note that if there is only one arc of positive curvature on the boundary, there is no second tallest undulation and the value of this feature is 0.

LoSmoothBndAngleTallestProcessesGFPCentroid, HiSmoothBndAngleTallestProcessesGFPCentroid: When there are two tallest processes, the angle (in degrees) subtended by the points on the unsmoothed cell boundaries that correspond to the points of maximum positive curvature of each of these two tallest smoothed boundary processes, relative to the cell segment's *GFP centroid*.

LoSmoothBndAngleTallestProcessesGFPCenterOfMass, HiSmoothBndAngleTallestProcessesGFPCenterOfMass: A feature variant of *LoSmoothBndAngleTallestProcessesGFPCentroid* (and its HiSmooth variant) where the cell center used for measuring the subtended angle is the *GFP center of mass*.

LoSmoothBndAngleTallestProcessesGFPBrightSpotCentroid, HiSmoothBndAngleTallestProcessesGFPBrightSpotCentroid: A feature variant of *LoSmoothBndAngleTallestProcessesGFPCentroid* (and its HiSmooth variant) where the cell center used for measuring the subtended angle is the *GFP bright spot centroid*.

LoSmoothBndLargestAreaForProcessGE0.5, HiSmoothBndLargestAreaForProcessGE0.5: The largest area of any process that has a maximum positive curvature >= 0.5.

LoSmoothBndLargestRelativeAreaForProcessGE0.5, HiSmoothBndLargestRelativeAreaForProcessGE0.5: The area of the process identified in *LoSmoothBndLargestRelativeAreaForProcessGE0.5* (and its HiSmooth variant), divided by the cell segment's total area.

LoSmoothBndSecondLargestAreaForProcessGE0.5, HiSmoothBndSecondLargestAreaForProcessGE0.5: The area of the process with *the second largest area*, for all processes with a maximum positive curvature >= 0.5. If there is no such process, the value of this feature is 0.

LoSmoothBndSecondLargestRelativeAreaForProcessGE0.5, HiSmoothBndSecondLargestRelativeAreaForProcessGE0.5: The area of the process identified in *LoSmoothBndSecondLargestAreaForProcessGE0.5* (and its HiSmooth variant), divided by the cell segment's total area.

LoSmoothBndAngleLargestProcessesGE0.5GFPCentroid, HiSmoothBndAngleLargestProcessesGE0.5GFPCentroid: When there are two processes with maximum positive curvature >= 0.5, the angle (in degrees) subtended by the points on the unsmoothed cell boundaries that correspond to the points of maximum positive curvature of each of these two largest smoothed boundary processes, relative to the cell segment's *GFP centroid*. Bipolar cells have a value of this feature that is close to 180. A value of 0 for this feature indicates that there is no second largest process.

LoSmoothBndAngleLargestProcessesGE0.5GFPCenterOfMass, HiSmoothBndAngleLargestProcessesGE0.5GFPCenterOfMass: A feature variant of *LoSmoothBndAngleLargestProcessesGE0.5GFPCentroid* (and its HiSmooth variant) where the cell center used for measuring the subtended angle is the *GFP center of mass*.

LoSmoothBndAngleLargestProcessesGE0.5GFPBrightSpotCentroid, HiSmoothBndAngleLargestProcessesGE0.5GFPBrightSpotCentroid: A feature variant of *LoSmoothBndAngleLargestProcessesGE0.5GFPCentroid* (and its HiSmooth variant) where the cell center used for measuring the subtended angle is the *GFP bright spot centroid*.

LoSmoothBndLargestAreaForProcessGE1, HiSmoothBndLargestAreaForProcessGE1: The largest area of any process that has a maximum positive curvature >= 1.

LoSmoothBndLargestRelativeAreaForProcessGE1, HiSmoothBndLargestRelativeAreaForProcessGE1: The area of the process identified in *LoSmoothBndLargestRelativeAreaForProcessGE1* (and its HiSmooth variant), divided by the cell segment's total area.

LoSmoothBndSecondLargestAreaForProcessGE1, HiSmoothBndSecondLargestAreaForProcessGE1: The area of the process with *the second largest area*, for all processes with a maximum positive curvature >= 1. If there is no such process, the value of this feature is 0.

LoSmoothBndSecondLargestRelativeAreaForProcessGE1, HiSmoothBndSecondLargestRelativeAreaForProcessGE1: The area of the process identified in *LoSmoothBndSecondLargestAreaForProcessGE1* (and its HiSmooth variant), divided by the cell segment's total area.

LoSmoothBndAngleLargestProcessesGE1GFPCentroid, HiSmoothBndAngleLargestProcessesGE1GFPCentroid: When there are two processes with maximum positive curvature >= 1, the angle (in degrees) subtended by the points on the unsmoothed cell boundaries that correspond to the points of maximum positive curvature of each of these two largest smoothed boundary processes, relative to the cell segment's *GFP centroid*. Bipolar cells have a value of this feature that is close to 180. A value of 0 for this feature indicates that there is no second largest process.

LoSmoothBndAngleLargestProcessesGE1GFPCenterOfMass, HiSmoothBndAngleLargestProcessesGE1GFPCenterOfMass: A feature variant of *LoSmoothBndAngleLargestProcessesGE1GFPCentroid* (and its HiSmooth variant) where the cell center used for measuring subtended angle is the *GFP center of mass*.

LoSmoothBndAngleLargestProcessesGE1GFPBrightSpotCentroid, HiSmoothBndAngleLargestProcessesGE1GFPBrightSpotCentroid: A feature variant of *LoSmoothBndAngleLargestProcessesGE1GFPCentroid* (and its HiSmooth variant) where the cell center used for measuring subtended angle is the *GFP bright spot centroid*.

Quantitative Morphological Analysis

Fisher Linear Discriminants (FLDs)

Initial work on classifier development employed a set of 502 cell segments comprising 11 TCs (including the GFP controls) and used Fisher Linear Discriminants. The composition of this set of cell segments is indicated in Table S3. It included RNAi knockdowns of Rac1, Rho1, and Cdc42 GTPases, as well as overexpression of activated forms of Rac1, Rho1, and the RhoGEF SIF1.

t-tests: We first computed t-test P values comparing the means of each feature in one of these TCs vs. the value of that feature in all other TCs combined (TC vs ~TC) and found that some individual features were strongly informative for some TCs. Indeed, 120 out of 1671 TC vs ~TC comparisons for single features were significant with P values < .05/1671. A TC where a N-terminally truncated form of SIF was overexpressed had the 8 most significant TC vs. ~TC P values for single features, with the DrainageArea feature exhibiting the lowest value of 1.2e-68. DrainageArea is a feature found to be informative for lamellipodia formation in a previous study (*1*). This finding is remarkably consistent with previous studies which have demonstrated that overexpression of N-terminally truncated forms of SIF stimulate lamellipodial protrusions (*10*). Moreover, 10 of the 11 TCs had at least one feature by which the t-test of the class against all other classes was significant at P<.05/1671.

Treatment Class (TC)	N	InFeat	Sens	Spec	OutFeat	Comment
ΔN-SIF	36	36	86	98	12	constitutively active SIF1
RhoGEF3_RNAi	26	42	73	98	8	RhoGEF3 knockdown
Gfp1	145	25	68	85	8	GFP controls
RhoV14	56	35	61	97	6	constitutively active Rho1
Rac1_Rac2_MTL_RNAi	21	10	24	100	3	triple knockdown
RacV12	45	26	24	98	6	constitutively active Rac1
Rho1_RNAi	33	21	9	100	7	Rho1 knockdown
control1	66	9	2	100	1	miscellaneous set of treatments
Dia_RNAi	19	8	0	100	1	Dia knockdown
Cdc42_RNAi	27					Cdc42 knockdown
Rac1_RNAi	28					Rac1 knockdown
TOTAL	502					

Table S3: Composition of set of cell segments used for initial classifier development and results of best Fisher linear discriminant (FLD) for each TC vs. ~TC derived as described in the text. N = number of cell segments in TC. InFeat = number of features after feature reduction (see text) considered by feature selection and discriminant construction logic. Sens = sensitivity of FLD (%) estimated from leave-one-out cross validation. Spec = specificity of FLD (%) estimated from leave-one-out cross-validation. OutFeat = number of features from InFeat set that were used in the best FLD. TCs are ordered from largest to least Sens. FLDs were constructed only for the first 9 of the 11 TCs in the 502 cell segments used for initial classifier development.

FLD construction: While t-tests showed that the means of some features were significantly different for a TC vs. all other TCs aggregated, well separated means do not imply well separated distributions. We therefore explored whether these features could be the basis of classifiers for predicting membership in a TC, starting with simple FLDs based on multiple informative features. Logic for building and evaluating discriminants was coded in MatLab and combined several processing steps including feature selection and leave-one-out cross-validation. Linear discriminants were developed to distinguish TC from ~TC for each of 9 of the TCs in the set of 502 cell segments.

Feature reduction: The feature set contains many sets of minor variants and contains other pairs of features which are highly correlated. To reduce redundancy, we first clustered features using complete linkage hierarchical clustering with a distance function of 1-abs(ρ), where ρ = the Pearson correlation coefficient for a pair of features over all 502 cell segments, and using a distance cutoff of 0.2 to define clusters, resulting in 82 clusters (including many singletons). Inspection verified that minor feature variants fell into the same clusters. We then assembled a reduced list of features for linear discriminant construction by selecting the feature from each cluster that had the lowest t-test P-value for TC vs. ~TC t-tests among all the features in the cluster, only retaining those whose P-value was less than a specified significance cutoff (.001 for all TCs except ΔN-SIF, for which it was .0001). This resulted in between 8 and 42 features available for FLD construction (see InFeat, Table S3).

FLD coefficient and threshold calculations: For a given non-singleton subset of the InFeat (Table S3) input features, FLD coefficients were computed in the usual way as $\mathbf{FLD_{coeff}} = (\mathbf{C_{TC}}+\mathbf{C_{\sim TC}})^{-1}\cdot(\mathbf{M_{TC}}-\mathbf{M_{\sim TC}})$, where $\mathbf{M_k}$ = the column vector of feature means for the class $\mathbf{k}$ ($\mathbf{k}$ = TC or ~TC), and $\mathbf{C_k}$ = the covariance matrix of the feature values over the N cell segments in TC (for $\mathbf{k}$ = TC), and over the 502-N cell segments in ~TC (for $\mathbf{k}$ = ~TC). Any cell segment s with feature values $\boldsymbol{f_s}$ can now be associated with a scalar value $(\mathbf{FLD_{coeff}})'\cdot \boldsymbol{f_s}$ and two classifiers constructed with any given threshold t, whereby for the first $(\mathbf{FLD_{coeff}})'\cdot \boldsymbol{f_s} < t$ assigns s to TC and $(\mathbf{FLD_{coeff}})'\cdot \boldsymbol{f_s} \geq t$ assigns s to ~TC, whereas for the second the class assignments are reversed, i.e., $(\mathbf{FLD_{coeff}})'\cdot \boldsymbol{f_s} < t$ assigns s to ~TC and $(\mathbf{FLD_{coeff}})'\cdot \boldsymbol{f_s} \geq t$ assigns s to TC. Finally, the value of the FLD threshold t and class assignment was chosen to minimize the average of the false positive and false negative error rates.

Feature selection: FLD construction was combined with feature selection using the Sequential Forward Floating Selection algorithm of (*20*). In this algorithm, one starts with a set comprising a single feature and proceeds through an unspecified number of steps each of which adds the feature outside of the current feature set that best improves performance, or subtracts the feature in the set that least degrades performance. The algorithm maintains a list of feature sets with 1, 2, 3, … features, and as it moves through addition and subtraction steps, it updates this list if a newly extended or subtracted set has better performance than the one currently in its list. When the algorithm ceases, the best performing set out of its list of feature sets with 1, 2, 3, … features defines the feature selection used for subsequent use of the classifier.

In our implementation, the singleton set used to start the algorithm consists of the feature with the best (lowest) TC vs. ~TC P-value, and the performance of the current set of features is measured by the average of the false positive and false negative error rates obtained from leave-one-out FLD construction and evaluation. This means that, to gauge performance of a feature set during any one iteration of the algorithm, 502 FLDs were constructed as described above with the current feature set -- one for each set of 501 cell segments excepting a hold-out case -- the hold-out cases are classified using their associated leave-one-out FLDs, and a false negative rate (FNR) and false positive error rates (FPR) is estimated based on these 502 classifications. The performance score used to evaluate algorithm iterations is the average of FPR and FNR.

The rationale for evaluating performance at each step using leave-one-out error is that, although very computationally expensive, this arrangement helps pick a feature set that has the best generalization properties rather than one that optimizes training set error, since errors and successes are judged for hold-out cases that are not used to construct the classifier used to evaluate them. The feature set constructed for an FLD is therefore also one in which generalization error is well controlled. Nevertheless, this strategy is limited not only by the limited search of feature sets supported by the algorithm of (*20*), but by the possibility of internal correlations within the TC cell segment sets, in which many cell segments are obtained from single wells.

The Table S3 OutFeat column presents the number of features selected by this algorithm for each of the 9 TCs for which FLDs were constructed.

Sensitivity and specificity: Sensitivity and specificity as given in Table S3 were also computed on the basis of leave-one-out cross-validations. E.g., false negative error for the ΔN-SIF FLD in Table S3 was evaluated as the FNR based on classifications in which one of the 36 ΔN-SIF TC cell segments was held out and classified from an FLD trained on the remaining 501 cell segments. Similarly specificity was computed from the FPR based on 466 ~ΔN-SIF hold-out FLDs and classifications.

	average FNR and FPR			**error rate *simpliciter***		
Treatment Class (TC)	**Sens**	**Spec**	**OutFeat**	**Sens**	**Spec**	**OutFeat**
ΔN-SIF	86	98	12	75	99	9
RhoGEF3 RNAi	73	98	8	65	67	4
Gfp1	68	85	8	50	51	2
RhoV14	61	97	6	64	65	7

Table S4: Comparison of sensitivity and specificity of Fisher Linear Discriminants constructed when error is computed as the average of false negative (FNR) and false positive error (FPR) (left) vs. error *simpliciter* in which negative and positive error is not distinguished (right). In general (but not always), use of error *simpliciter* results in reduced sensitivity. See text for details.

A note on error evaluation: We note that the choice of evaluating performance by *averaging FPR and FNR* has the effect of compensating for the inequality between the sizes of the positive and negative training sets -- e.g. 36 positive vs. 466 negative cases for ΔN-SIF. We found that if performance were judged on error rate *simpliciter* -- i.e., the total fraction of misclassifications regardless of whether the misclassification were for a positive or negative case -- the resulting classifiers and feature sets tended to be ones in which specificity was emphasized at the expense of sensitivity, presumably because when negative cases dominate, false negative error dominates the total error. This is seen in Table S4, which compares results for the first four TCs and FLDs in Table S3. Although, statistically, use of error rate *simpliciter* is frequently a valid strategy, it resulted here in classifiers that tended to predict very few positive cases, and for that reason we employed the average of FPR and FNR. As can be seen from Table S4, however, this failed to produce sensitive FLDs for several TCs, whose sensitivities remained very low while their specificities were near 100%.

Principal Component Analysis (PCA): We also explored use of PCA for analyzing the discriminative power of multiple features. For instance, Figure S13 plots ΔN-SIF vs. ~ΔN-SIF samples using the first three principal components based on the 12 features selected for the optimal ΔN-SIF vs, ~ΔN-SIF FLD above.

In general we did not find PCA particularly informative with these data. A key issue is that the linear combinations of features that represent principal components could not be easily interpreted compared to the original features presented clearly meaningful properties of cell images. Another issue is that PCA, which is based on variance and covariance calculations, could be affected by outliers (e.g., note that there are several points on the extreme left in Figure S13).

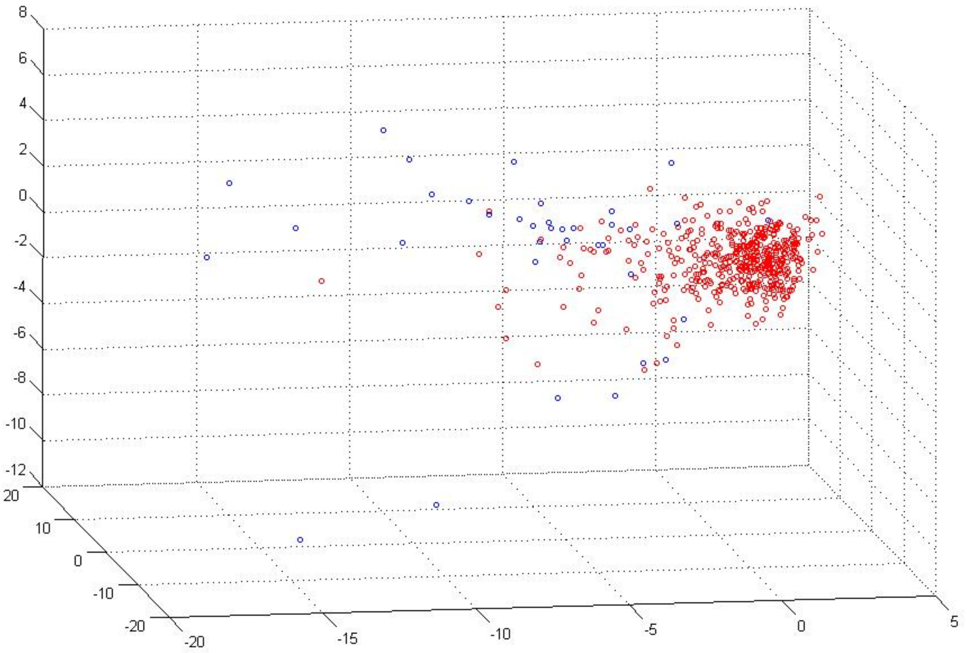


Figure S13: ΔN-SIF (blue circles) and ~ΔN-SIF (red circles) cell segments as plotted for the first three principal components for the 12 features for the optimal ΔN-SIF vs. ~ΔN-SIF FLD (see text). The blue circles are mainly on the left.

Fisher Linear Discriminants -- Conclusions: Many instances were found in which individual features generated by automated image analysis for stochastically labeled cells were highly informative concerning TC membership, as judged by t-tests against the mean feature values of these TC. Modest success was achieved in classifying some TCs against the rest by using multiple features in simple FLDs. The best classifier, for the ΔN-SIF TC, achieved 86% sensitivity and 98% specificity. Effective FLDs could not be constructed for several TCs, however, and these tended to have very low sensitivities and close to perfect specificities. In some cases, this may have been due to the training set itself, as this contained instances of TCs that were probably very similar and would therefore be difficult to distinguish using FLDs: For example, the TCs Rac1_RNAi and Rac1_Rac2_MTL_RNAi both involve use of the same Rac1 dsRNA which generated cells with similar phenotypes that would have appeared as both positive and negative cases during the construction of the FLD for Rac1_Rac2_MIT_RNAi vs. ~Rac1_Rac2_MTL_RNAi.

Another observation is that the November 2005 GFP TC (GFP control set) yielded a relatively poor FLD with only 68% sensitivity. A similar result was obtained for the more powerful neural network classifiers (below). We speculate that non-control TCs are perturbed by expression constructs or RNAi knockdowns that serve to drive phenotype in certain determinate ways that can support construction of effective classifiers, while, in contrast, the gfp1 TC is not driven in any particular manner and so is less able to support such classifiers. This has implications for the image analysis strategy of defining classifiers for normal cells as a way of automating identification of abnormal cells, for it suggests that normal cells may be more intrinsically variable and thus less able to support such classifiers.

Neural Networks

Our promising but modest results with Fisher linear discriminants (FLDs) encouraged us to explore more powerful classifiers. We opted to move to Neural Networks (NNs) in favor of other classification strategies such as Support Vector Machines (SVMs) because of the convenience of being able to continue working in MatLab using the Neural Network Toolkit, and because SVMs operate on linear combinations of transformed feature values we believed would be hard to interpret based on our experience with principal component analysis (above). We based our strategy for building NNs on a variant of the procedures used for FLDs, but the higher computational demands of NN training required considerable streamlining. We used MatLab 2006b and Neural Network toolkit version 5.0.1. We summarize our procedures below:

We emphasize also that while we developed and trained NNs in their capacity as classifiers, our principal interest in and usage of them was to use NN scores as quantitative measures of similarity of cells to treatment classes. Thus in our final analysis we nowhere actually use NNs to classify cells, but only use them to assess whether cells are morphologically similar to the cells in a target TC.

Data set for NN training: Table S5 describes the composition of the set of 804 cell segments (S804) used for NN training. This set contains 9 TCs that feature additional constitutively active alleles for Rac1 and Rho1, the ΔN-SIF TC described earlier, a TC where the full-length form of the RhoGEF SIF is overexpressed (vs. ΔN-SIF TC where a N-terminally truncated form of SIF is overexpressed), and overexpression constructs for two additional GEFs: CG3799 and ΔNRhoGEF3. Unlike the initial attempts to derive classifiers described previously where TCs were selected based on the particular gene that was targeted by RNAi or overexpressed, all TCs (except for gfp1) used for NN-based methods were selected due to the fact that cells of each TC were qualitatively distinct from control cells.

Treatment Class	N	Qualitative Phenotype
CG3799_overexp	55	Long, bipolar shaped cells. Large cell body.
ΔNRhoGEF3_const_overexp	57	Very small, often perfectly round cells with little variation between cells.
gfp1	145	Polar cells with leading-edge lamellipodia, filipodia, and trailing edge.
ΔN-SIF	36	Very large cells with extensive lamellipodia. Loss of polarity
RacF28L	130	Extensive lamellipodia and filopodia formation. Loss of polarity
RacV12	45	Extensive lamellipodia formation. Loss of polarity.
RhoF30L	128	Small compacted cells with jagged, “fuzzy” edges.
RhoV14	56	Extensive, long and irregular protrusions. Cells body appears small and retracted.
SIF1_full_overexp	152	Largely appear wild-type
TOTAL	804	

Table S5: Composition of set of cell segments used for NN training

Classifiers developed: NN classifiers were trained for each of the 9 TC vs ~TC comparisons for individual TCs. However, we also trained NNs for three additional comparisons involving pairs of similar TCs; in particular, we trained NNs for

- RacV12 and RacF28L (175 cell segments) vs. the aggregate of all other TCs (631 cell segments); these are distinct constitutively active Rac1 alleles. The set of 175 positive cases was denoted RacV12_RacF28L.
- RhoV14 and RhoF30L (184 cell segments) vs. the aggregate of all other TCs (620 cell segments); again, these are distinct consitutively active Rho1 alleles. The set of 184 cell positive cases was denoted RhoV14_RhoF30L.
- ΔN-SIF and SIF1_full_overexp (188 cell segments) vs. the aggregate of all other TCs (652 cell segments); ΔN-SIF is a constitutively active form of SIF1, whereas SIF1_full_overexp is a constitutively expressed full length wild type SIF1 gene. The set of 188 positive cases was denoted ΔN-SIF_SIF1_full_overxp.

Because positive sets for NN training are not always TCs (treatment classes) but sometimes unions of TCs, we use the more general term "target class" (TGC) to describe the positive cases of our NN training procedures.

Feature reduction and reduction: Features were clustered, t-tests for TGC vs. ~TGC comparisons were performed, and the best (i.e., lowest t-test P-value) feature from every cluster selected, in a manner identical to the way these steps were performed for FLDs, yielding a ranked reduced list of features. However, unlike for FLDs, no cutoff on P-values was used to further reduce features. Instead, because NN training is much more computationally intensive than FLDs, and because it requires consideration of many network architectures, we used a much simpler method of feature selection than the already computationally intensive implementation of the Sequential Forward Floating

Selection algorithm we used for FLDs. Instead, when training a NN whose architecture involved *k* input features, we simply used the first *k* features from this ranked list. The NN architectures we used were such that $1 \le k \le 12$ (see *NN architectures*).

NN architectures: We trained and tested the set of 33 small and simple network architectures described in Figure S14. These networks all involved small numbers of input (up to 12, see *Feature selection and reduction* above) and intermediate neurons to limit the serious potential for overtraining entailed by our small training sets.

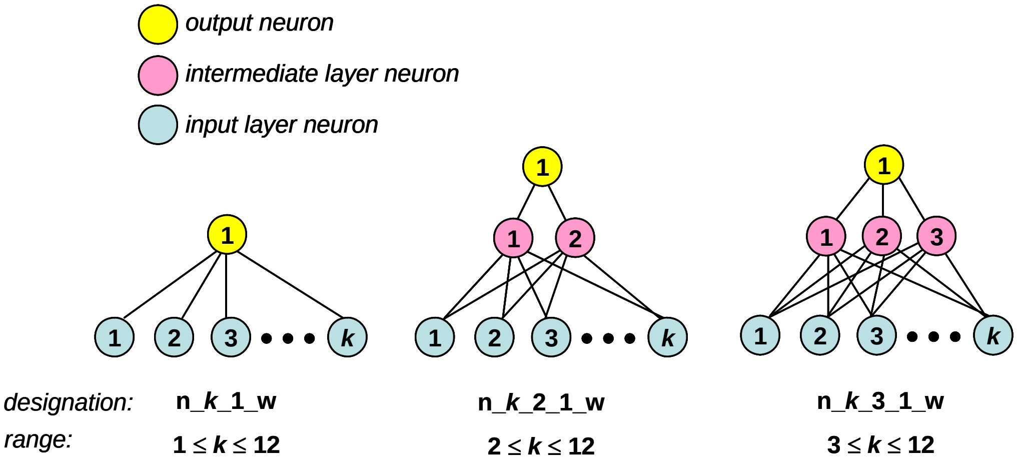


Figure S14: Neural network architectures trained and tested for TGC vs. ~TGC classifier development. Note the designations used to describe each architecture.

NN training parameters, classification: NNs were defined to use log-sigmoid transfer functions for every intermediate and output neuron. Optimization used default MatLab parameters except for explicit specification of 250 epochs, a training function of 'trainscg' (scaled conjugate gradient backpropagation), and use of NN error weights described below. Target values for training were 1 for all positive (TGC) cases and 0 for all negative (~TGC) cases. After training, cases were classified as positives (TGC) if their NN score ≥ 0.5 and negative otherwise.

NN error weights: Preliminary testing with NNs indicated that, similar to the case of FLDs (above), sensitivity of NNs was improved when NNs were optimized during training to minimize *average false positive and false negative error* vs. total error. To achieve this weighing, a custom "performance function" (i.e., error calculation function) was written in MatLab that computed error as in Equation (1).

$$(1) \qquad error = \frac{1}{2} \cdot \left(\frac{1}{|Pos|} \cdot \sum_{i \in Pos} \left(NNScore_i - 1 \right)^2 + \frac{1}{|Neg|} \cdot \sum_{i \in Neg} NNScore_i^{\,2} \right)$$

In this equation, *Pos* = the set of positive (TGT) cases, *Neg* = the set of negative (~TGT) cases, and $NNscore_i$ is the score for case *i* computed by the neural network. Error computed by the default MatLab performance function is unweighted in that it does not employ the factors 1/|*Pos*| and 1/|*Neg*|, or the coefficient 1/2. Our error calculation procedures is thus analogous to our use of the average of the FPR and FNR used in FLDs above, except that square differences of NN scores from training targets are used instead of counts of false positives and false negative errors. An example of how this weighting improved sensitivity is shown in Figure S15.

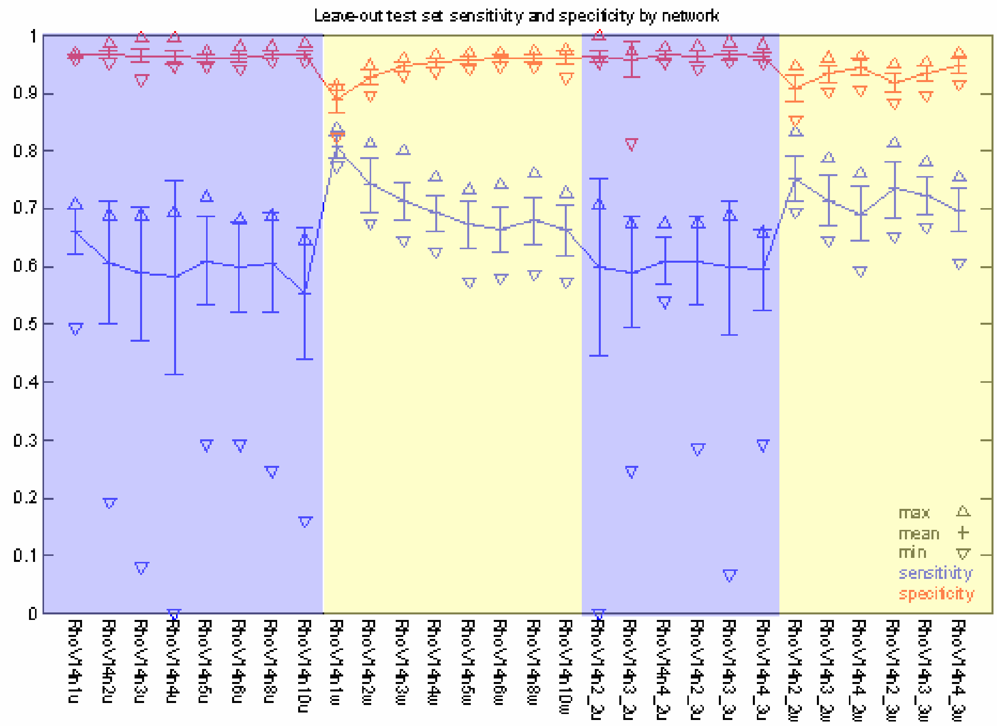


Figure S15: Example of effect of using error weights described in equation (1) from preliminary results collected in the process of developing the neural network training algorithms described in the text. The graph shows the sensitivity and specificity estimated on multiple leave out test sets of neural networks trained for the RhoV14 target class (TGC) using a variant of procedures described in the text. Along the abscissa are NN architectures, where those ending in a "w" were trained using error weights (1) and those ending in a "u" were unweighted and did not use (1). Regions of the graph in blue are "u", regions in yellow are "w", and the architectures in each blue area are identical to and in the same order as the architectures in the yellow areas to their right. Sensitivity (lower line) is generally higher for "w" training than "u" training for corresponding architectures. Specificity (upper line) is generally lower for "w" training than "u" training. These results indicate that use of error weights (1) achieves substantial improvements in sensitivity at the cost of some loss of specificity.

NN initial weights and biases: The non-linear optimization required to train an NN is affected by initial seed values for neuron connection weights and biases (IWB). In MatLab's implementation, IWBs are chosen randomly if not explicitly specified, and preliminary testing indicated that the IWB could have substantial influence on the success of NN training (see Figure S16). Therefore, for each NN architecture considered, we captured 25 sets of MatLab-generated random IWBs, and used and re-used these in all

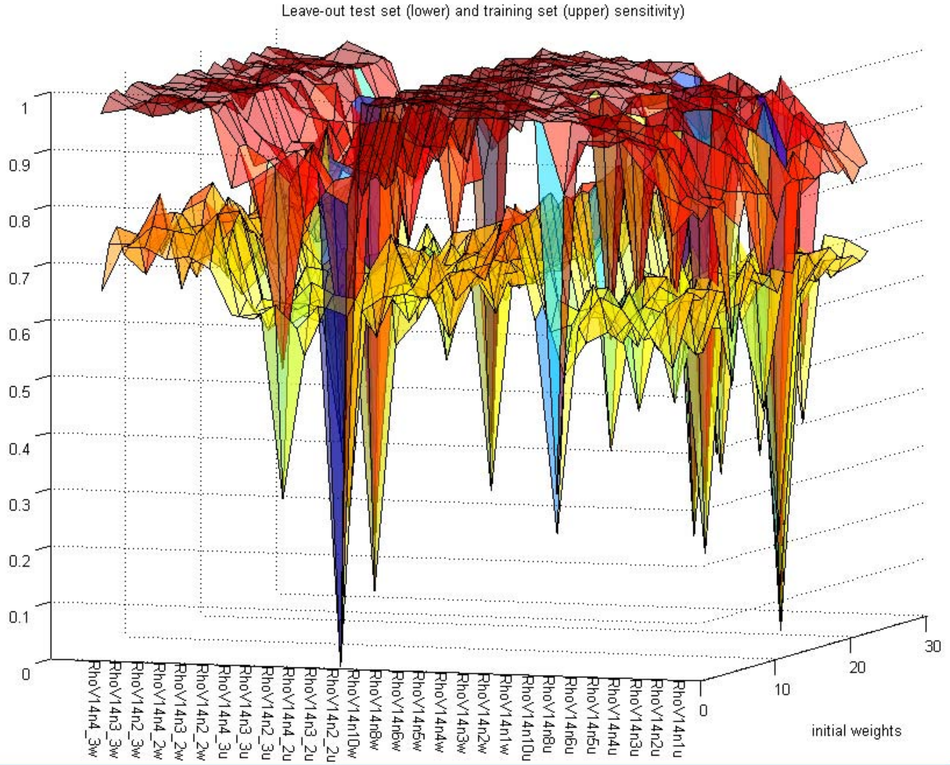


Figure S16: Effect of random initial weights and biases on NN sensitivity. Shown are the test set (lower surface) and training set sensitivity (upper surface) for each of the neural network architectures and 25 random NN initial weight and bias sets, averaged over the random test / training sets, from the same runs as featured in Figure S15. The initial random weight and bias set exhibits substantial influence over the success of a given NN architecture as measured by sensitivity.

evaluations of the architecture. We designed NN training to choose not only the best NN architecture, but also the best of its 25 associated IWBs.

NN training: The algorithm used to train and select the best NN is diagrammed in Figure S17. For each of our 12 TGCs, we performed a preliminary screening of our 33 NN architectures, in which we assessed the success of each of the NN architectures on a set of 25 random partitions of the S804 segment set into test and training sets, and 25 IWBs. In each of the 25 random partitions, 10% of the positive and 10% of the negative cases in S804 were chosen as the test set, and the remaining 90% were used for training. All NN training used the NN error weights described above. The success of each NN architecture was assessed by means of a *distance measure* (DM) defined as follows: The average test set specificity and average test set sensitivity (averaged over the 625 trainings of the architecture [i.e., 25 IWBs × 25 training/test sets], and expressed as fractions between 0 and 1) were plotted as a (x,y) coordinates on a plane, and the Euclidean distance between this point and the point (1,1) was calculated. The DM therefore assesses how far the NN architecture, when trained, deviates from perfect test set sensitivity and specificity.

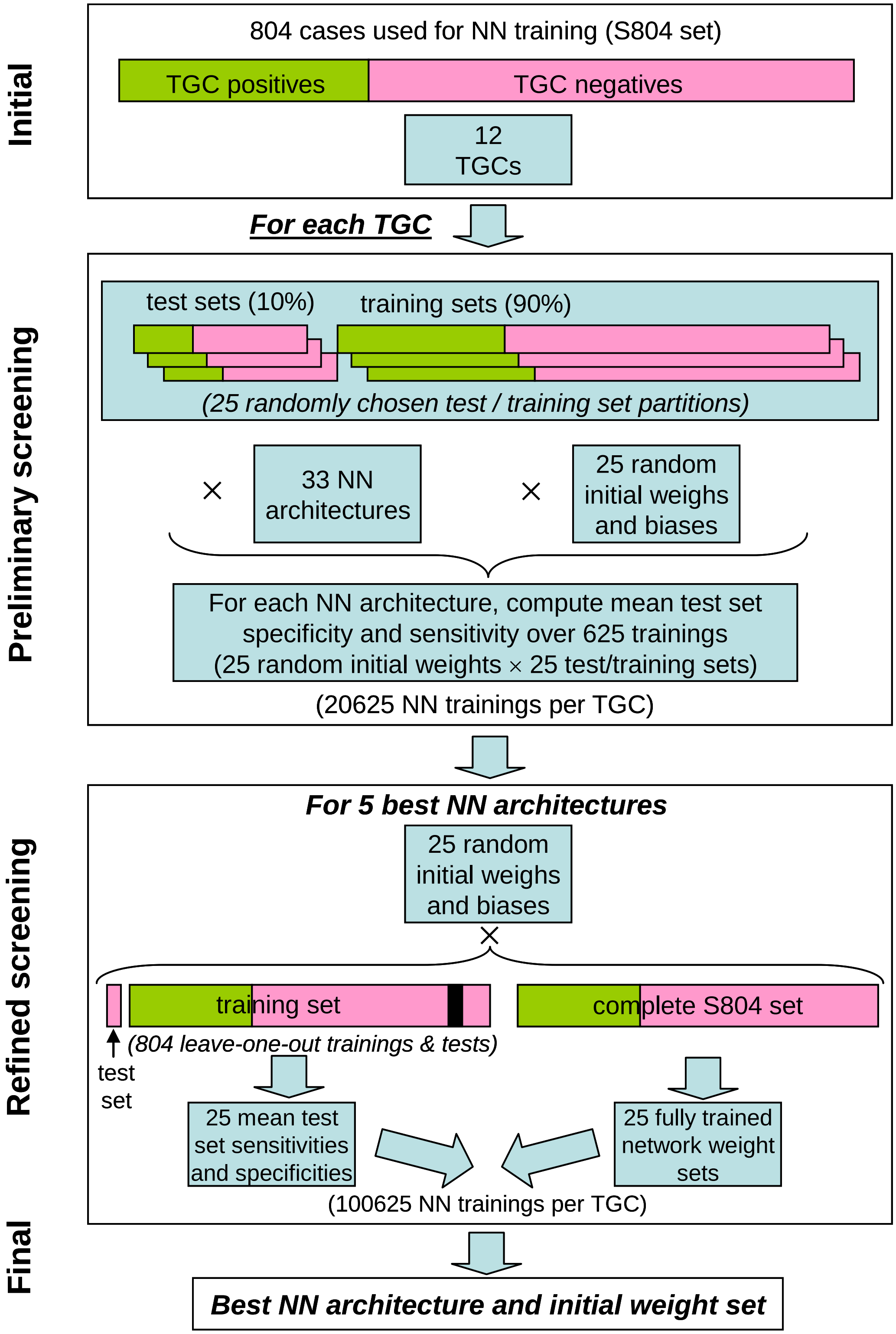

Initial
804 cases used for NN training (S804 set)
TGC positives
TGC negatives
12 TGCs
For each TGC
Preliminary screening
test sets (10%)
training sets (90%)
(25 randomly chosen test / training set partitions)
×
33 NN architectures
×
25 random initial weighs and biases
For each NN architecture, compute mean test set specificity and sensitivity over 625 trainings
(25 random initial weights × 25 test/training sets)
(20625 NN trainings per TGC)
Refined screening
For 5 best NN architectures
25 random initial weighs and biases
×
training set
complete S804 set
(804 leave-one-out trainings & tests)
test set
25 mean test set sensitivities and specificities
25 fully trained network weight sets
(100625 NN trainings per TGC)
Final
Best NN architecture and initial weight set

Figure S17: Schematic of NN training algorithm (see text for details).

The five NN architectures that exhibited the five lowest DMs in preliminary screening were deemed the five best architectures, and these were then tested more thoroughly in a subsequent refined screening step: Each of these architectures was subjected to comprehensive leave-one-out testing with the S804 segment set for each of the 25 IWBs. The architecture and IWB among these 5 × 25 combinations with the lowest DM was then selected as the best NN architecture and IWB. Final network weights and biases were computed for this NN architecture from its associated IWB and trained on the complete S804 set, and the resulting weights and biases then used to score all cell segments in our complete data set with this NN.

		Final NN							Prelim NNs	
		S804		Set of 5383 segments		S804 leave-1-out				
Target Class (TGC)	Final NN	Sens	Spec	Sens	Spec	Mean Sens	Mean Spec	DM	Best NN rank	Best Prelim NN
ΔNRhoGEF3_const_overexp	N_12_2_1_w	100.0	95.6	100.0	94.4	100.0	95.7	0.04	2	n_12_1_w
ΔN-SIF	N_5_1_w	97.2	93.1	97.2	93.8	97.2	92.7	0.08	1	n_5_1_w
RhoV14	N_3_1_w	89.3	85.7	89.3	76.9	85.7	85.7	0.20	1	n_3_1_w
RhoF30L	N_9_3_1_w	92.2	84.3	92.2	77.3	86.7	84.2	0.21	4	n_9_2_1_w
RacF28L	N_12_1_w	89.2	81.6	89.2	76.4	86.2	81.0	0.24	3	n_10_1_w
RacV12_RacF28L	N_9_1_w	81.1	81.1	81.1	78.0	86.3	80.8	0.24	4	n_5_1_w
CG3799_overexp	N_9_1_w	81.8	87.2	81.8	83.7	76.4	87.0	0.27	3	n_10_1_w
RhoV14_RhoF30L	N_11_3_1_w	88.0	82.4	88.0	70.7	82.1	79.8	0.27	1	n_11_3_1_w
RacV12	N_12_1_w	88.9	75.8	88.9	75.1	82.2	75.0	0.31	1	n_12_1_w
gfp1 (GFP controls)	N_11_2_1_w	80.7	62.8	80.7	46.8	66.9	72.2	0.43	5	n_10_1_w
ΔN-SIF_SIF1_full_overexp	N_10_1_w	71.3	68.0	71.3	63.3	71.3	67.4	0.43	1	n_10_1_w
SIF1_full_overexp	N_10_1_w	71.1	64.9	71.1	58.4	69.1	64.3	0.47	1	n_10_1_w

Table S6: Characteristics of NNs derived from training algorithm described in Figure S17 and text. **Best NN** = NN architecture that emerged as best after training. **Final NN** = the best NN architecture trained on the entire S804 segment set using the best initial weights and biases (IWB) determined by the training algorithm. Sensitivity (**Sens**) and specificity (**Spec**) are provided for the Final NN as computed on the entire S804 set and a larger set of 5383 segments comprising the first 12 batches of cell segments processed for features (see above). Mean sensitivity (**Mean Sens**) and specificity (**Mean Spec**) computed for the Final NN architecture and IWB from the 804 leave-1-out trainings and tests in the refined screening of the NN training algorithm (see Figure S17) are also presented, as well as the distance measure **DM** (see text). All sensitivities and specificities are given as percentages. TGCs are listed in increasing DM order, i.e., from best to the worst performing NN. In most cases Final NN S804 Sens and Spec exceed or equal the Mean Sens and Mean Spec from the S804 leave-1-out testing, a possible result of training of the Final NN on the complete S804 set vs. the 803 segments used for training in S804 leave-1-out training. As described in Figure S17, the Best NN was chosen from one of the five best NNs determined from preliminary screening. The **Best Prelim NN** (as measured by DM) is indicated, as is the rank of the Final NN among the 5 preliminary NN architectures. In half (six) of the TGCs, the Final NN is different from the Best Prelim NN, indicating that the refined screening of the NN training algorithm (Figure S17) was effective in choosing a better performing architecture. In five of these six cases (yellow background), the Final NN architecture was more complex or used a greater number of input features than the Best Prelim NN, indicating that refinement depended on increasing the number of features and and/or the complexity of feature processing. In the single exception (pink background), refinement involved removing one feature.

For each TGC, this NN training algorithm performed 121250 NN trainings, so that a total of 1455000 trainings were performed over our entire set of 12 TGCs. Characteristics of the best NNs for each TGC are given in Table S6. NN data including the final trained network's weights and biases and the NN's input features are presented on our supplementary web site http://arep.med.harvard.edu/QMS in the form of MatLab statements that can be (and were) used to recover the final trained NNs and compute scores for all cell segments in our data set. Classifier scores generated for all 12,601 cell segments are provided as supplemental files accessible from this same web site.

NN development -- Conclusions: NNs generated classifiers with excellent performance for some of our TGCs, including cells treated with our ΔNRhoGEF3 overexpression, ΔN-SIF, and RhoV14 constructs. As noted above, a principal reason for working with NNs was to develop classifiers with better sensitivity compared to FLDs. A comparison of Tables S3 and S5 for ΔN-SIF, RhoV14, and gfp1 (GFP controls) shows that NNs did, in fact, exhibit improved sensitivity, and this was also true of NNs for Rac1-related constructs (FLD data not shown in Table S4). As improved sensitivity frequently comes at the cost of lower specificity, specificities for the NN classifiers are generally modest but still comparable or better than corresponding FLDs. We emphasize that these sensitivities and specificities describe the error of the NN classifiers on the task of classifying a single cell segment, and (as is generally true of image analysis), the performance of the classifiers would be expected to be improved when classifying entire TCs comprising multiple similar cells -- and this was the context in which we used these NNs (see section on **Quantitative Morphological Signatures** and **Clustering and Replicability Analysis** below). As noted above, we used NN scores as similarity scores indicating the degree to which a TC was like one of our TGCs, rather than as tools for actual classification of cell segments.

Similar to the case of FLDs, the NN developed for gfp1 (November 2005 GFP controls) had relatively poor performance, consistent with our speculation above that cells that are not driven by a construct to adopt particular morphologies are more variable morphologically and thus less amenable to successful classifier development on the basis of shape and content. Perhaps surprisingly, in view of the good performance of the ΔN-SIF NN, the NN trained for a full SIF1 overexpression construct performed very poorly. This may indicate either that ΔN-SIF overexpression affect cell morphology in a more robust fashion than overexpression of full-length SIF. Such a model is consistent with the prediction that ΔN-SIF is an activated RhoGEF, whereas the majority of expressed full-length SIF is presumably in an autoinhibited conformation (*7*).

In the case of NNs based on Rac1 and Rho1 constructs, we found that NNs were capable of distinguishing constructs based on different alleles of the same gene with good performance. However the possibility that different samples of essentially similar cells are subject to sample and treatment effects that induce enough difference in morphology for a NN to distinguish them should not be excluded. However, when evaluating whole TCs comprising multiple cells, and combining multiple NNs in the context of clustering, we find evidence of replicability of our NN-based analysis (see **Clustering and Replicability Analysis** below) and conclude that both of these steps help improve the statistical power of our NNs.

These replicability results, however, were based only on a subset of the better performing NNs. The NN for combination TGC RhoV14_RhoF30L performed less well than either of the individual RhoV14 and RhoF30L NNs, while that for RacV12_RacF28L had performance in between the corresponding individual NNs. Again, this result is ambiguous between the hypothesis that the individual TCs within the TGCs are more different than they are similar, and that the larger and more variable combination TGCs could only support inferior classifiers. We therefore dropped the combination TGCs from consideration and selected individual Rac1- and Rho1- based NNs for further use, and also dropped the gfp1 and SIF1_full_overexp NNs from consideration for the reasons cited above.

A final observation is that several of the NNs involve a single layer of input neurons connecting to a single output neuron, an architecture that should divide feature space with a single separation plane in a manner equivalent to FLDs; however we judge the performance of the NNs to be superior to that of the corresponding FLDs in the cases of ΔN-SIF, RhoV14, and gfp1. Why did we not find FLDs as good as these NNs? We note that our judgment of 'better performance' is weighted towards better sensitivity, and that although the error weighting we used for NNs vs our FLDs is similar, the frameworks are not identical, so that one possibility is simply that the error weighting used for NNs was more successful for our goal of improved sensitivity. However, it is also true that unlike these NNs, FLDs are constrained by the Fisher heuristic to find a separation plane perpendicular to a specific line in feature space; therefore the NNs have more freedom to find a better separation. Finally, several of our best NNs do employ intermediate layer neurons and therefore find better separations by employing separators more complex than planes -- that these architectures were not chosen as optimal for several of our TGC vs. ~TGC classifications suggests that their inherently better performance was compromised by overtraining that was detected and rejected by our use of leave-out cross-validation in our NN training algorithm. The fact that several of our NNs reduce to planar separators may therefore indicate only that, given the small size of our data sets, overtraining could only be avoided by adopting the most simple NN architectures.

Quantitative Morphological Signatures (QMSes)

While our feature analysis generated 145 numerical features corresponding to aspects of morphology for each cell segment and thus provided immense information about cell morphology, these features had complex relations to each other and unclear biological interpretation. We used our NN classifiers to see past these internal relationships and provide biological interpretation by means of QMSes.

For each TC in our data set, we defined a QMS as a vector of NN *Z scores* for the 7 most successful our NNs -- ΔN-SIF, ΔNRhoGEF3_const_overexp, CG3799_const_overexp, RhoV14, RhoF30L, RacV12, and RacF30L. Specifically, for a given TC and NN

$$NN\ Z\ score = \frac{\mu_{NN,TC} - \mu_{NN,all}}{\sigma_{NN,all} / \sqrt{|TC|}}$$

where $\mu_{NN,all}$ and $\sigma_{NN,all}$ are the mean and standard deviation of the scores for the NN classifier over all 12,601 segments. The NN Z score thus measures the difference between the mean NN score for the TC and the mean of the NN score over all segments, compared to the standard deviation of the mean of a set equal in size to the TC of randomly chosen segments. NN Z scores thus provide a measure of the degree of confidence that a TC is different from random on the scale defined by that NN. Since the TGC used to train an NN has scores near the maximum possible score of 1 for an NN, a high NN Z score for a TC indicates that the TC rates high on this NN compared to random cell segments and is thus similar in morphology to the morphology of the TGC. We used NN Z scores to define QMSes instead of the NN scores themselves to compensate for the difference in statistical power of the NNs trained on different TGCs, described above. In short, a QMS for a TC describes the similarity of the morphology of that TC relative to the reference classes ΔN-SIF, ΔNRhoGEF3_const_overexp, CG3799_const_overexp, RhoV14, RhoF30L, RacV12, and RacF30L. It thus uses our feature analysis to provide a biological interpretation of the TC's feature information in a convenient quantitative form.

Clustering and Replicability Analysis

We used QMSes to cluster the 273 TCs in our data set. Hierarchical average linkage clustering was performed using Cluster and TreeView (*21*) using uncentered Pearson Correlation Coefficients as the distance measure, and individual clusters were defined interactively by finding the highest nodes at which the distance measure became greater than .8. Several thresholds were evaluated and this threshold was chosen because this level of correlation resulted in coherent groups of qualitatively similar cells, meaning that clusters determined by smaller correlation scores included cells that were visually morphologically diverse, whereas determining clusters based on higher correlation scores resulted in visually morphologically similar cells becoming segregated into distinct groups. Clustering was performed multiple times during the course of our data gathering and analysis.

We used clusters generated from a set of 11312 segments that represented the first 13 of the 14 batches of cell segments that we gathered over the course of this analysis. These segments contained 32 examples of treatments that were repeated in 2, 3, or 5 distinct samples (see Table S2). In 15/32 cases, 2-5 different amplicons were used to target the same gene, while in 16/32 cases the same amplicon was present in different wells (1/32 was a control well with no dsRNA). We clustered these TCs by their QMSes using the procedures above, maintaining distinct identities for the individual repeated treatments, so that each individual sample within a set of replicates found its own individual place in the clustering, resulting in a partition of 273 TCs (with replicates distinguished) across 13 clusters. We computed the average number of distinct clusters occupied by the distinct replicate TCs for treatments with 2, 3, and 5 replicates, a value we called the *co-clustering index*. We then randomized cluster assignments 5000 times to compute a null distribution of co-clustering index, and estimated the P value of finding the actual-coclustering index. The results are shown in Table S7.

RepsetSize	**2**	**3**	**5**	
RepsetCount	29	2	1	
ActualCoClus	1.58621	2	1	
AvgRandCoClus	1.75132	2.3308	3.2014	mean(CoClus) over all random clusterings
StdRandCoClus	0.079684	0.438531	0.825454	std(CoClus) over all random clusterings
MinRandCoClus	1.37931	1	1	min(CoClus) over all random clusterings
Pvalue	**0.0368**	0.409	**0.007**	Fraction of random clusterings with CoClus <= actual CoClus

Table S7: Replicability test statistics for clustering of treatments with replicates distinguished (see text for details)

For the 29 cases of 2 replicates, and for the single case of 5 replicates, P<.05 (bold) provides evidence that actual replicates co-cluster better than random clusterings. This result was not obtained for the case of three replicates, although this involved only two cases. We took these results as indicating that our feature analysis, NN scoring, QMS-based clustering produced results that were robust to biological replicates, although in the case of two replicates, the fact that the P value was close to .05 suggests substantial variability across replicates. We stress that many of these replicates were biological and not technical replicates, which would increase the degree of variability.

On the basis of these results, we combined all replicate treatment segment samples into the same TCs in all further analysis. The final clustering of all 12,601 cell segments across 249 TCs yielded 41 clusters. Cluster assignments for each TC are presented below in Table S8.

Enrichment Statistics

We computed enrichment statistics for our final clustering against functional category information derived from GeneOntology (*22*), results of prior screens reported in the literature, and results of prior screens in our laboratory (www.flyrnai.org and flight.licr.org), using standard hypergeometric statistics, i.e.,

$$P_{C,F} = \sum_{x=O(C,F)}^{\min(|C|,|F|)} \frac{\binom{|C|}{x}\binom{N-|C|}{|F|-x}}{\binom{N}{|F|}}$$

where C = a cluster, F = a functional category, $O(C,F)$ = number of TCs that are in common between C and F (i.e, the overlap), N = total number of TCs that both appear in the clustering and which have functional category assignments, and $|C|$ and $|F|$ represent the sizes of C and F, respectively. Calculations were performed using the MatLab hygecdf function. In the case of screens, F was treated as $\{0,1\}$, where 1 = the condition of passing the screen and 0 = the condition of not passing the screen.

While many $P_{C,F} < .05$ were found, the number of statistical tests was so large that none of these P values were statistically significant after correcting for multiple hypotheses, even restricting attention to comparison of the clustering to a single set of functional categories. Nevertheless, a marginally significant P value of 5.78E-05 was computed for the enrichment of cluster 33 (19 members) by genes annotated with the function RhoGAP (17 genes represented in the $N = 249$ TCs in the overall clustering), which had an overlap of 7 genes. This involved 1271 statistical tests of 41 clusters with 31 functions, implying a Bonferroni-corrected critical value of .05/1271 = 3.9E-05.

Nevertheless, we used these P-values to describe *relative* functional enrichment in clusters, simply by assigning the function with the lowest P-value as the "most enriched" function in a cluster. Based on concordant literature and other experimental observations cited in the text, we feel these enriched classes may be biologically relevant (see text). In some cases additional observations concerning these P-values supported these conclusions: For example, the P value of 0.0245698 was computed for the enrichment of cluster 6 by genes annotated as Rap signaling components (3/5 genes in dataset), whereas no other cluster had a P<0.8 value for Rap components.

Supplemental Text

Here we discuss in detail additional findings of our study, and provide further validation for our experimental and statistical methods.

Adhesion turnover is intimately coupled with the regulation of cell body retraction and the stimulation of cortical tension.

In addition to being enriched in ArfGAPs, Cluster 18 contains dsRNAs targeting *Gα49B, Gαι65A, Gβ13F*, the Gα-subunit *concertina,* and the *loco* gene that encodes an RGS-containing protein (Figure S18). *Concertina* has repeatedly been implicated in promoting cortical tension by acting as an upstream activator of Rho/ROCK/myosin activity, both in tissue culture and whole organism models (*23-25*). Interestingly, overexpression of *Gαι65A* in BG-2 cells results in highly-rounded cells with few protrusions that cluster with ΔN-RhoGEF3 overexpressing cells (Cluster 3). We propose that *Gαι65A* and other heterotrimeric G-proteins and regulators within this cluster likely function in a modular signaling cassette that promotes cortical tension. Inhibiting the activity of this cassette results in a general loss of tension and deregulated cell spreading that is quantitatively similar to inhibition of adhesion turnover pathways. We propose that adhesion disassembly and the upregulation of cortical tension must be regulated simultaneously, and we have identified a local network of proteins which regulates these activities. In fact mammalian GIT1, which is involved in adhesion disassembly was originally isolated through its physical interactions with a G-protein receptor coupled kinase (*26*), which

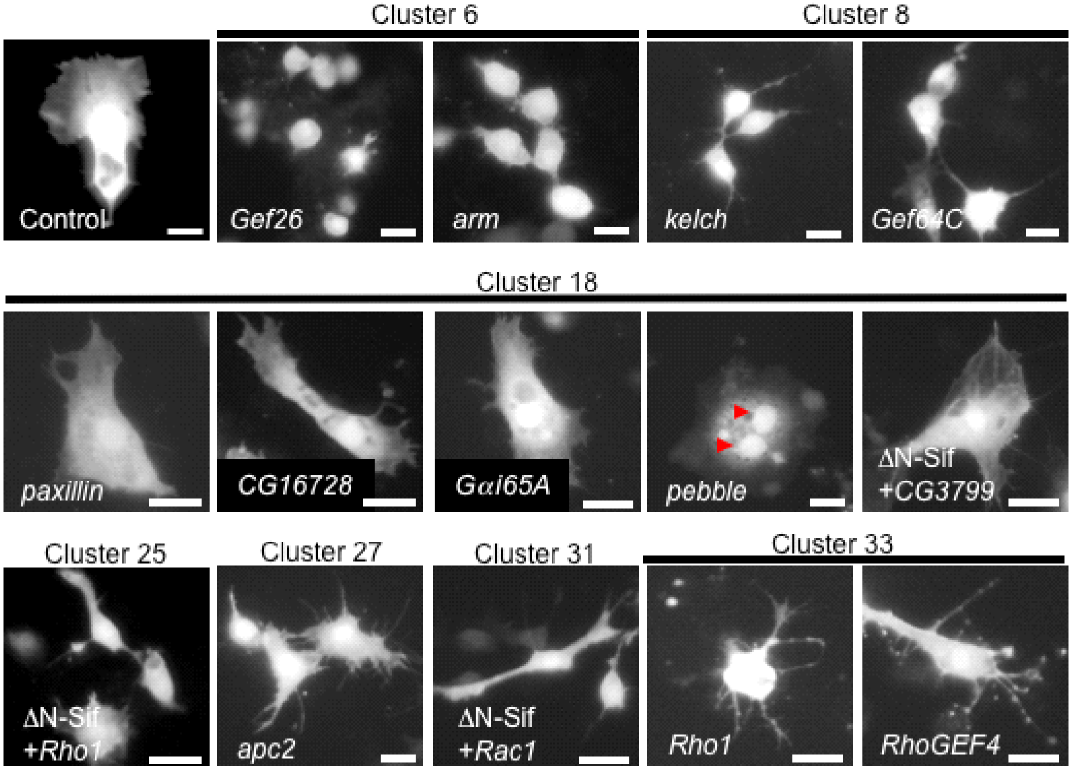


Figure S18. Phenoclusters Represent Functionally Related Genes. (A) Control BG-2 cells (expressing GFP) have a polarized leading-edge and a retracted tailing-edge. *Gef26* and *armadillo* RNAi results in loss of cellular protrusions (Cluster 6). *Kelch and GEF64C* RNAi cells display protrusive activity, but have few or poorly-formed lamellipodia. *Paxillin*, *CG16728, pebble*, and *Gια65A* RNAi are members of the same phenocluster where cells are extensively spread and/or have large protrusions (Cluster 18), but Pebble-deficient cells are typically multi-nucleated (chevrons) suggesting these cells do not undergo cytokinesis. Cluster 18 also contains cells overexpressing ΔN-SIF in combination with *CG3799* dsRNA that are indistinguishable from cells expressing ΔN-SIF alone. However, *Rac1* or *Rho1* dsRNAs in combination with ΔN-SIF overexpression results in phenotypes that fall into different phenoclusters (Clusters 25 and Cluster 31 respectively). Apc-2 deficient cells display an aberrant number of long protrusions (Cluster 27). *Rho1* and *RhoGEF4* dsRNA are members of the same phenocluster (Cluster 33). All scale bars equal 10 μM.

suggests ArfGAP signaling, such as through *CG16728*, acts to couple the upregulation of tension with adhesion turnover.

QMSes for *pebble* and *RacGAP50C* dsRNA, also co-cluster with *paxillin* and ArfGAP dsRNA QMSses (Cluster 18). Both *pebble* and *RacGAP50C* have well-established roles in the regulation of cytokinesis. Visual inspection of cells where *pebble* and *RacGAP50C* have been targeted by dsRNA reveals that while these cells are phenotypically similar to other cells in the cluster, they are also bi-/multi-nucleated strongly suggesting that normal cytokinesis has been inhibited resulting in their large/spread and unpolarized morphology (Figure S18). Thus, currently our methods do not have the power to distinguish failures in cytokinesis from defects in adhesion turnover and cortical tension, but we do not exclude the possibility all these processes may be regulated by similar regulatory pathways that involve Pebble and RacGAP50C.

Morphologies indistinguishable from wild-type cells

A cluster of genes that is defined by QMSes with high RhoV14 and CG3799 NNZs includes both RhoV14 overexpression, *CG3799* overexpression and RNAi, GFP1 and GFP2, and TCs where a number of different cDNAs are overexpressed (Cluster 33). However given our ability to generate two unique classifiers for both RhoV14-, and CG3799-, expressing cells, the changes in cell shape that RhoV14 and CG3799 promote are statistically distinct from each other, as well as from the shape of control cells. Thus in some cases our current methods do not have to power to resolve particular phenotypes or signaling states even when qualitative and quantitative differences may exist amongst these phenotypes.

QMSes as Readouts of Signaling Pathway Activity

We reasoned that if our assay indeed captures the roles of specific genes/protein in signaling pathways that control cell morphology, modifying or perturbing the activity of these genes/proteins through both direct and indirect means should be reflected in their QMSes. Therefore we tested the effects of a limited set of dsRNAs for their ability to suppress the phenotype induced by overexpression of ΔN-SIF. dsRNAs targeting *CG799, Rab5, RhoGAP16F, RhoGAP54D, and MTL* failed to suppress the effects of ΔN-SIF overexpression, as cells expressing ΔN-SIF and simultaneously treated with these dsRNA have QMSes that co-cluster with the QMS of ΔN-SIF+*GFP* dsRNA (Cluster 18). *RhoGEF3* dsRNA only mildly suppressed the effects of ΔN-SIF overexpression (Cluster 17). However the addition of dsRNAs targeting *Rac1* (Cluster 31), *Rho1* (Cluster 25), *enabled* (Cluster 33), and *Arc-p34* (Cluster 33) significantly altered the QMS of ΔN-SIF expressing cells, suggesting that the proteins encoded by these genes act downstream of SIF activity. This is consistent with the observation that both SIF and Tiam-1 can act as Rac-specific GEFs, and that Rho activity is required for effect of Tiam-1 on mammalian cell morphology (*27*). Furthermore, Tiam-1 directly binds the Arp2/3 complex of which Arc-p34 is a member (*28*). Taken together these results suggest that quantitative morphological signatures can serve as sensitive readouts for both cell morphology and the activity of signaling pathways, and highlight the fact that signaling pathways can be modeled using phenotypic data.

QMSes are predictive of biochemical activity

Rho1 dsRNA clusters with a small group of genes that includes *RhoGEF2* dsRNA, a well-characterized direct and specific upstream activator of Rho1 GTPase activity (*29-32*) (Cluster 33). Therefore, we reasoned that RhoGEF4, an uncharacterized GEF that when inhibited results in a QMS that is part of this cluster, may also be a direct activator of Rho1. Using an *in vitro* method that detects total levels of active Rho in cell lysates, we observe that dsRNA targeting of *RhoGEF4* specifically reduces total Rho activity in BG-2 cells, whereas dsRNAs against a variety of other *RhoGEFs* do not (Figure S19). These data support the notion that RhoGEF4, like RhoGEF2, is a major upstream regulator of Rho activity in BG-2 cells. Furthermore, dsRNAs targeting *RhoGEF3* and

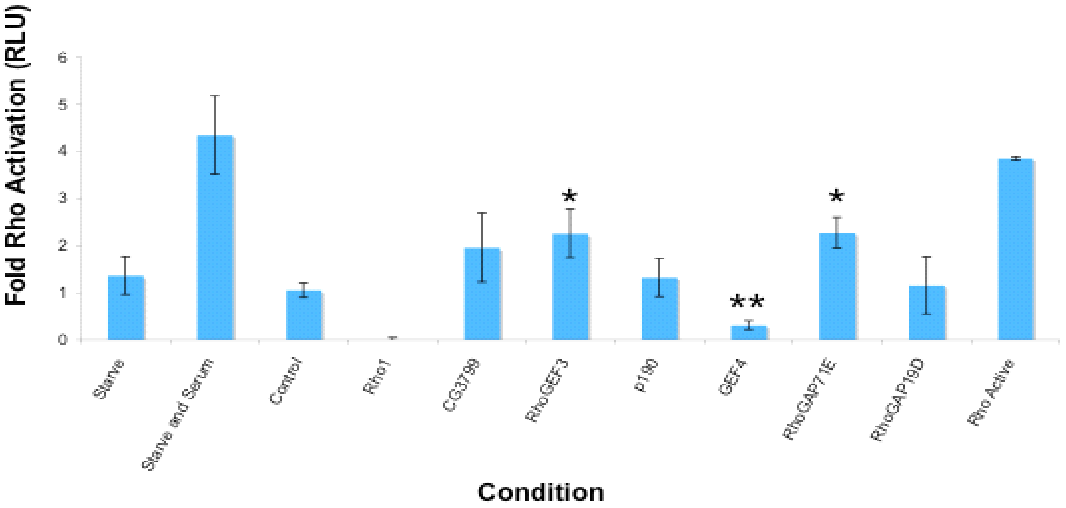


Figure S19. *In vitro* Rho activation assay determining total Rho activity in BG-2 cell extracts prepared as follows: control/GFP-alone transfections; *Rho1* dsRNA; serum-starved cells (24 hrs); serum-starved (24 hrs) cells stimulated with serum for 30 min; *CG3799* dsRNA; *RhoGEF3* dsRNA cells; *p190RhoGAP* dsRNA; *RhoGEF4* dsRNA; *RhoGAP71E* dsRNA; and *RhoGAP19D* dsRNA. As a control we also monitored Rho activation using purified recombinant RhoQ60L. *RhoGEF4* dsRNA resulted in a significant decrease in total Rho activation (P=0.003), while *RhoGEF3* dsRNA (P=0.017), and *RhoGAP71E* dsRNA (P=0.014) resulted in significant increases in total Rho activation. Each data point represents the normalized mean value of independent experiment where N>3. Error bars represent the mean +/- SD. Fold Rho activation corresponds to relative luciferase activity versus that of control experiment. Significance was determined using Student's T-test. A single asterisk denotes P<0.05, two asterisks denotes P<0.01.

*RhoGAP71*E increased basal Rho activity in BG-2 cells (Figure S19), which is remarkably consistent with the fact that inhibition of these genes results in a QMSes with high RhoF30L NNZs. Taken together, these data suggest that quantitative morphological profiling can be used to simultaneously monitor signaling activity on both local (i.e. at sites of adhesion assembly or actin polymerization) and global/cell-wide scales.

Supplemental Tables

Table S8: Assignment of treatment conditions to particular phenoclusters based on clustering of QMSes with Correlation Distance Cutoff > 0.80

Treatment Class	Gene	Symbol	Pheno Cluster	Pathway	Function
Rac1	FBgn0010333	Rac1	1	Rho	Rho GTPase
GEF64C overexpression			1	Rho	RhoGEF
Rac1Rac2MTL			1	Rho	Rho GTPase
hAuroraB const. active overexpression			1		
CG32627	FBgn0052627	CG32627	2		
G protein beta 76C	FBgn0004623	G[beta]76C	3	G-protein	G-gamma
Microtubule-associated protein 205	FBgn0002645		4		
CG8707	FBgn0033272	CG8707	4	Rag	Rag GTPase
RhoGAP71E	FBgn0036518	RhoGAP71E	4	Rho	RhoGAP
sec23	FBgn0037357	sec23	4		
GXIVsPLA2	FBgn0036545	GXIVsPLA2	5		
CG7940	FBgn0038576	CG7940	5		
armadillo	FBgn0000117	arm	6		
Centrosomal protein 190kD	FBgn0000283	Cp190	6		
l(1)dd4	FBgn0001612	l(1)dd4	6		
Sop2	FBgn0001961	Sop2	6		
lightoid	FBgn0002567	ltd	6	Rab	Rab GTPase
staufen	FBgn0003520	stau	6		
slingshot	FBgn0003971	shot	6		
Ankyrin	FBgn0011747	Ank	6		
Arf51F	FBgn0013750	Arf51F	6	Arf	Arf GTPase
Merlin	FBgn0013951	Mer	6		
rho-like	FBgn0014380	RhoL	6	Rho	Rho GTPase
Gef26	FBgn0021873	Gef26	6	Rap	RapGEF
cib	FBgn0026084	cib	6		
C3G	FBgn0026145	C3G	6	Rap	RapGEF
RhoGAPp190	FBgn0026375	RhoGAPp190	6	Rho	RhoGAP
CG8801	FBgn0028473	CG8801	6		
CG7578	FBgn0028538	CG7578	6	Arf	ArfGEF
CG1583	FBgn0030013	CG1583	6		
CG9699	FBgn0030772	CG9699	6	Septin	Septin GTPase
CG7846	FBgn0030877	CG7846	6		
CG4267	FBgn0031405	CG4267	6		
Rab30	FBgn0031882	Rab30	6	Rab	Rab GTPase
CG5160	FBgn0031906	CG5160	6		
CG5337	FBgn0032249	CG5337	6	Rab	TBC GTPase

CG9426	FBgn0032485	CG9426	6		
Rab9	FBgn0032782	Rab9	6	Rab	Rab GTPase
CG9248	FBgn0032923	CG9248	6		
CG4853	FBgn0034230	CG4853	6	Ras	RasGEF
CG10540	FBgn0034577	CG10540	6		
RhoGEF3	FBgn0035128	RhoGEF3	6	Rho	RhoGEF
CG33232	FBgn0035347	CG33232	6		
CG6838	FBgn0037182	CG6838	6		
CG4448	FBgn0039067	CG4448	6		
SCAR	FBgn0041781	SCAR	6	Rho	Rho effector
RapGAP1	FBgn0053529	Rapgap1	6	Rap	RapGAP
G protein alpha-i 65A overexpression			6	G-protein	G-beta
del-N-RhoGEF3 overexpression			6	Rho	RhoGEF
yurt	FBgn0004049	yrt	7		
Rab-protein 3	FBgn0005586	Rab3	7	Rab	Rab GTPase
Neurofibromin 1	FBgn0015269	Nf1	7	Ras	RasGAP
CG6017	FBgn0036555	CG6017	7		
CG1193	FBgn0037375	CG1193	7		
kelch	FBgn0001301	kel	8		
RanGAP	FBgn0003346	RanGap	8	Ran	RanGAP
alpha-Catenin	FBgn0010215	[alpha]-Cat	8		
twinstar	FBgn0011726	tsr	8		
cnn	FBgn0013765	cnn	8		
Septin-2	FBgn0014029	Septin-2	8	Septin	Septin GTPase
Trio	FBgn0024277	trio	8	Rho	RhoGEF
Grip75	FBgn0026431	Grip75	8		
Rab3-GAP	FBgn0027505	rab3-GAP	8	Rab	Rab GTPase
capt	FBgn0028388	capt	8		
CSN1a	FBgn0028838	CSN1a	8		
CG3009	FBgn0029720	CG3009	8		
Marf	FBgn0029870	Marf	8		
Graf	FBgn0030685	Graf	8	Rho	RhoGAP
Rab35	FBgn0031090	Rab35	8	Rab	Rab GTPase
Arc-p20	FBgn0031781	Arc-p20	8		
gartenzwerg	FBgn0033714	garz	8	Arf	ArfGEF
CG15611	FBgn0034194	CG15611	8	Rho	RhoGEF
Mapmodulin	FBgn0034282	Mapmodulin	8		
CG15097	FBgn0034396	CG15097	8		
GEF64C	FBgn0035574	Gef64C	8	Rho	RhoGEF
RhoGAP68F	FBgn0036257	RhoGAP68F	8	Rho	RhoGAP
Rab26	FBgn0037072	Rab26	8	Rab	Rab GTPase
CG32030	FBgn0052030	CG32030	8		
RhoF30L overexpression			8	Rho	Rho GTPase
Mtl	FBgn0039532	Mtl	9	Rho	Rho GTPase

Vav	FBgn0040068	vav	10	Rho	RhoGEF
CG5745	FBgn0038855	CG5745	11	Rab	TBC GTPase
Cdep	FBgn0032821	CdGAPr	12	Rho	RhoGEF
CG7324	FBgn0037074	CG7324	13	Rab	RabGAP
del-N-SIF overexpression			14	Rho	RhoGEF
canoe	FBgn0000340	cno	15	Rap	Rap Effector
ran-like	FBgn0036497	ran-like	16	Ran	Ran GTPase
Microtubule-associated protein 60	FBgn0010342	Map60	17		
miranda	FBgn0021776	mira	17		
jitterbug	FBgn0028371	jbug	17		
p16-ARC	FBgn0031437	p16-ARC	17		
del-N-SIF_RhoGEF3dsRNA			17		
concertina	FBgn0000384	cta	18	G-protein	G-alpha
Actinin	FBgn0000667	Actn	18		
G protein alpha-i 65A	FBgn0001104	G-i[alpha]65A	18	G-protein	G-beta
G protein beta 13F	FBgn0001105	G[beta]13F	18	G-protein	G-gamma
Mp20	FBgn0002789	Mp20	18		
pbl	FBgn0003041	pbl	18	Rho	RhoGEF
gamma-tubulin at 23C	FBgn0004176	[gamma]Tub23C	18		
G protein alpha 49B	FBgn0004435	G[alpha]49B	18	G-protein	G-alpha
gamma tubulin 37C	FBgn0010097	[gamma]Tub37C	18		
Gelsolin	FBgn0010225	Gel	18		
ADP ribosylation factor 79F	FBgn0010348	Arf79F	18	Arf	Arf GTPase
Moesin	FBgn0011661	Moe	18		
sanpodo	FBgn0011716	spdo	18		
Actin-related protein 66B	FBgn0011744	Arp66B	18		
Rab-protein 2	FBgn0014009	Rab2	18	Rab	Rab GTPase
Rab5	FBgn0014010	Rab5	18	Rab	Rab GTPase
Rab-RP4	FBgn0015794	Rab-RP4	18	Rab	Rab GTPase
locomotion defects	FBgn0020278	loco	18		
Phospholipase A2 activator protein	FBgn0024314	Plap	18		
CG14782	FBgn0025381	CG14782	18		
Brahma associated protein 55kD	FBgn0025716	Bap55	18		
Rap21	FBgn0025806	Rap2l	18	Rap	Rap GTPase
Crag	FBgn0025864	Crag	18		
Septin-5	FBgn0026361	Septin-5	18	Septin	Septin GTPase
Grip84	FBgn0026430	Grip84	18		
mini spindles	FBgn0027948	msps	18		
falten	FBgn0028380	fal	18		
centaurin gamma 1A	FBgn0028509	cenG1A	18	Arf	ArfGAP
alpha-catenin related	FBgn0029105	alpha-catenin-related	18		
Patsas	FBgn0029137	Patsas	18		

lava lamp	FBgn0029688	lva	18		
pod1	FBgn0029903	pod1	18		
CG12102	FBgn0030180	CG12102	18		
CG11063	FBgn0030530	CG11063	18		
RhoGAP15B	FBgn0030808	RhoGAP15B	18	Rho	RhoGAP
RhoGAP19D	FBgn0031118	RhoGAP19D	18	Rho	RhoGAP
CG13692	FBgn0031254	CG13692	18	Arf	ArfGAP
CG9135/RCC-1	FBgn0031769	CG9135	18	Ran	RanGEF
Menin 1	FBgn0031885	Mnn1	18		
Arc-p34	FBgn0032859	Arc-p34	18		
CG9243	FBgn0032926	CG9243	18		
Grp1	FBgn0032960	Grp1	18	Arf	ArfGEF
CG12736	FBgn0033184	CG12736	18		
CG16728	FBgn0033539	CG16728	18	Arf	ArfGAP
Dystrobrevin-like	FBgn0033739	Dyb	18		
RacGAP50C	FBgn0033881	RacGAP50C	18	Rho	RhoGAP
CG8479	FBgn0033914	CG8479	18	Dynamin	Dynamin GTPase
CG5522	FBgn0034158	CG5522	18	Ral	Ral GEF
CG15609	FBgn0034180	CG15609	18		
RhoGAP54D	FBgn0034249	RhoGAP54D	18	Rho	RhoGAP
EfSec	FBgn0034627	EfSec	18		
CG33275	FBgn0035802	CG33275	18	Rho	RhoGEF
CG10971	FBgn0036309	CG10971	18		
CG10724	FBgn0036357	Aip1	18		
CG7365	FBgn0036939	CG7365	18		
Sar1	FBgn0038947	sar1	18	Sar	Sar GTPase
cenB1A	FBgn0039056	cenB1A	18	Arf	ArfGAP
RhoGAP100F	FBgn0039883	RhoGAP100F	18	Rho	RhoGAP
paxillin	FBgn0041789	Pax	18		
CG18858	FBgn0042175	CG18858	18		
CG30158	FBgn0050158	CG30158	18		
CG30440	FBgn0050440	CG30440	18	Rho	RhoGEF
CG30456	FBgn0050456	CG30456	18	Rho	RhoGEF
CG31683	FBgn0051683	CG31683	18		
MICAL	FBgn0053208	MICAL	18		
RacF28L overexpression			18	Rho	Rho GTPase
RacV12 overexpression			18	Rho	Rho GTPase
SIF full-length overexpression			18	Rho	RhoGEF
del-N-SIF + GFP dsRNA			18		
del-N-SIF_CG3799dsRNA			18		
del-N-SIF_Rab5dsRNA			18		
del-N-SIF_RhoGAP16FdsRNA			18		
del-N-SIF_RhoGAP54D dsRNA			18		
del-N-SIF+MTL_RNAi			18		

homolog of RecQ	FBgn0027375	RecQ5	19		
SIF	FBgn0019652	SIF	20	Rho	RhoGEF
G protein gamma 30A	FBgn0028433	G[gamma]30A	20	G-protein	G-gamma
CG5022	FBgn0032225	CG5022	20		
RhoBTB	FBgn0036980	RhoBTB	20	Rho	Rho GTPase
formin 3	FBgn0053556	form3	20	G-protein	G-alpha
enabled	FBgn0000578	ena	21	Rho	Rho effector
CG14034	FBgn0031691	CG14034	22		
CG8243	FBgn0033349	CG8243	23	Arf	ArfGAP
cappuccino	FBgn0000256	capu	24	Rho	Rho Effector
G protein s-alpha 60A	FBgn0001123	G-s[alpha]60A	24	G-protein	G-gamma
shibire	FBgn0003392	shi	24	Dynamin	Dynamin GTPase
capping protein beta	FBgn0011570	cpb	24		
CG7420	FBgn0031344	CG7420	24	Ran	RanGEF
mbc	FBgn0015513	mbc	25	Rho	RhoGEF
CG7787	FBgn0032020	CG7787	25	Rab	RabGEF
Rheb	FBgn0041191	Rheb	25	Rheb	Rheb GTPase
del-N-SIF+Rho1_RNAi			25		
RabX2	FBgn0030200	RabX2	26	Rab	Rab GTPase
CG14045	FBgn0040387	CG14045	26		
dia	FBgn0011202	dia	27		
apc	FBgn0015589	Apc	27		
apc2	FBgn0026598	Apc2	27		
Fimbrin	FBgn0024238	Fim	28	G-protein	G-alpha
CG32138	FBgn0052138	CG32138	28	Rho	Rho Effector
Muscle-specific protein 300	FBgn0053715	Msp-300	28		
chrowded	FBgn0015372	chrw	29		
CG8557	FBgn0030842	CG8557	30	Rho	RhoGEF
CG10188	FBgn0032796	CG10188	30	Rho	RhoGEF
Elongation factor 1?48D	FBgn0000556	Ef1[alpha]48D	31		
no receptor potential A	FBgn0004625	norpA	31		
Cdc42	FBgn0010341	Cdc42	31	Rho	Rho GTPase
rtGEF	FBgn0015803	rtGEF	31	Rho	RhoGEF
par1	FBgn0026193	par-1	31		
CG3799	FBgn0027593	CG3799	31	Rho	RhoGEF
CG11490	FBgn0031233	CG11490	31	Rab	TBC GTPase
Pld	FBgn0033075	Pld	31		
CG11968	FBgn0037647	CG11968	31	Rag	Rag GTPase
CG12241	FBgn0038304	CG12241	31	Rab	TBC GTPase
RhoGAP92B	FBgn0038747	RhoGAP92B	31	Rho	RhoGAP
CG30115	FBgn0050115	CG30115	31	Rho	RhoGEF
Moody beta overexpression			31	G-protein	GPCR
Cdc42Y32A overexpression			31	Rho	Rho GTPase
CG3799 overexpression			31	Rho	RhoGEF
RhoV14			31	Rho	Rho GTPase

del-N-SIF+Rac1_RNAi			31		
dLis1 overexpression			31		
dPar1 overexpression			31		
dStrad overexpression			31		
gfp1			31		
gfp2			31		
Nrg overexpression			31		
TumL overexpression			31		
visceral mesodermal armadillo-repeats	FBgn0022960	vimar	32		
G protein o-alpha 47A	FBgn0001122	G-o[alpha]47A	33	G-protein	G-gamma
Sos	FBgn0001965	Sos	33	Rho	RhoGEF
alpha-Spectrin	FBgn0003470	[alpha]-Spec	33		
Rho1	FBgn0014020	Rho1	33	Rho	Rho GTPase
RhoGEF2	FBgn0023172	RhoGEF2	33	Rho	RhoGEF
RhoGAP1A	FBgn0025836	RhoGAP1A	33	Rho	RhoGAP
RhoGAP5A	FBgn0029778	RhoGAP5A	33	Rho	RhoGAP
RhoGAP16F	FBgn0030893	RhoGAP16F	33	Rho	RhoGAP
RhoGAP18B	FBgn0030986	RhoGAP18B	33	Rho	RhoGAP
RhoGEF4	FBgn0035761	RhoGEF4	33	Rho	Rho GTPase
CG7323	FBgn0036943	CG7323	33	Rho	RhoGEF
RhoGAP93B/Vilse	FBgn0038853	RhoGAP93B	33	Rho	RhoGAP
RhoGAP102A	FBgn0039898	RhoGAP102A	33	Rho	RhoGAP
CG30372	FBgn0050372	CG30372	33	Arf	ArfGAP
RabX4	FBgn0051118	RabX4	33	Rab	Rab GTPase
CdGAPr	FBgn0051536	Cdep	33	Rho	RhoGAP
del-N-SIF_Arcp34dsRNA			33		
del-N-SIF_EnadsRNA			33		
dMemo overexpression			33		
Rab-protein 7	FBgn0015795	Rab7	34	Rab	Rab GTPase
peanut	FBgn0013726	pnut	35		
CG7197	FBgn0035866	CG7197	36	Arf	Arf GTPase
Bj1 protein	FBgn0002638	Bj1	37		
CLIP-190	FBgn0020503	CLIP-190	37		
CG14507	FBgn0039655	CG14507	37		
Rab-protein 6	FBgn0015797	Rab6	38	Rab	Rab GTPase
CG8397	FBgn0034066	CG8397	39		
G protein gamma 1+A27	FBgn0004921	G[gamma]1	40		
abnormal spindle	FBgn0000140	asp	41		

Table S9: dsRNA amplicons used in this study and predicted number of off targets. See http://www.flyrnai.org for sequence information.

Gene	Symbol	Amplicon	19 bp OT
FBgn0010215	[alpha]-Cat	DRSC11917	0
FBgn0003470	[alpha]-Spec	DRSC08704	0
FBgn0004176	[gamma]Tub23C	DRSC00820	1
FBgn0010097	[gamma]Tub37C	DRSC03535	1
FBgn0000667	Actn	DRSC17724	1
FBgn0036357	Aip1	DRSC09787	0
FBgn0029105	alpha-catenin-related	DRSC04669	2
FBgn0011747	Ank	DRSC17127	0
FBgn0015589	Apc	DRSC14114	0
FBgn0026598	Apc2	DRSC14115	0
FBgn0031781	Arc-p20	DRSC02917	2
FBgn0032859	Arc-p34	DRSC02113	0
FBgn0013750	Arf51F	DRSC05921	0
FBgn0010348	Arf79F	DRSC11606	0
FBgn0000117	arm	DRSC18738	0
FBgn0011744	Arp66B	DRSC09669	2
FBgn0000140	asp	DRSC16903	0
FBgn0025716	Bap55	DRSC07000	0
FBgn0002638	Bj1	DRSC09684	2
FBgn0026145	C3G	DRSC22329	463
FBgn0028388	capt	DRSC03331	0
FBgn0000256	capu	DRSC00434	0
FBgn0010341	Cdc42	DRSC20228	2
FBgn0051536	Cdep	DRSC12220	0
FBgn0032821	CdGAPr	DRSC03289	0
FBgn0039056	cenB1A	DRSC16923	1
FBgn0028509	cenG1A	DRSC02538	1
FBgn0028509	cenG1A	DRSC03505	0
FBgn0032796	CG10188	DRSC02023	0
FBgn0034577	CG10540	DRSC04080	0
FBgn0036309	CG10971	DRSC09812	0
FBgn0030530	CG11063	DRSC19367	3
FBgn0031233	CG11490	DRSC00313	1
FBgn0037375	CG1193	DRSC12184	4
FBgn0037647	CG11968	DRSC14450	0
FBgn0030180	CG12102	DRSC17804	0
FBgn0030180	CG12102	DRSC22159	0
FBgn0038304	CG12241	DRSC14481	1

FBgn0033184	CG12736	DRSC06163	0
FBgn0031254	CG13692	DRSC00361	0
FBgn0031691	CG14034	DRSC02359	0
FBgn0040387	CG14045	DRSC18600	0
FBgn0039655	CG14507	DRSC14866	2
FBgn0025381	CG14782	DRSC18568	0
FBgn0034396	CG15097	DRSC06526	1
FBgn0034180	CG15609	DRSC06561	1
FBgn0034194	CG15611	DRSC06563	0
FBgn0034194	CG15611	DRSC06564	1
FBgn0030013	CG1583	DRSC18094	1
FBgn0033539	CG16728	DRSC06620	1
FBgn0042175	CG18858	DRSC02703	0
FBgn0029720	CG3009	DRSC18263	1
FBgn0050115	CG30115	DRSC06748	0
FBgn0050158	CG30158	DRSC04974	0
FBgn0050158	CG30158	DRSC05012	0
FBgn0050372	CG30372	DRSC06493	0
FBgn0050372	CG30372	DRSC06830	0
FBgn0050440	CG30440	DRSC04845	0
FBgn0050456	CG30456	DRSC06565	2
FBgn0051683	CG31683	DRSC02703	0
FBgn0052030	CG32030	DRSC10545	0
FBgn0052138	CG32138	DRSC10716	13
FBgn0052627	CG32627	DRSC19404	0
FBgn0035347	CG33232	DRSC08187	1
FBgn0053167	CG33275	DRSC10866	0
FBgn0027593	CG3799	DRSC09673	7
FBgn0031405	CG4267	DRSC00647	1
FBgn0039067	CG4448	DRSC22351	1
FBgn0034230	CG4853	DRSC06898	1
FBgn0032225	CG5022	DRSC02802	0
FBgn0031906	CG5160	DRSC02821	1
FBgn0032249	CG5337	DRSC02841	0
FBgn0034158	CG5522	DRSC06941	1
FBgn0038855	CG5745	DRSC15867	0
FBgn0036555	CG6017	DRSC10570	0
FBgn0037182	CG6838	DRSC11794	7
FBgn0035866	CG7197	DRSC10785	3
FBgn0036943	CG7323	DRSC29701	0
FBgn0037074	CG7324	DRSC11820	0
FBgn0036939	CG7365	DRSC10824	1

FBgn0031344	CG7420	DRSC00704	0
FBgn0028538	CG7578	DRSC01893	0
FBgn0032020	CG7787	DRSC03069	0
FBgn0030877	CG7846	DRSC20059	1
FBgn0038576	CG7940	DRSC16350	1
FBgn0033349	CG8243	DRSC07122	0
FBgn0033349	CG8243	DRSC22608	67
FBgn0034066	CG8397	DRSC07162	0
FBgn0033914	CG8479	DRSC07191	1
FBgn0030842	CG8557	DRSC20111	2
FBgn0033272	CG8707	DRSC07239	0
FBgn0028473	CG8801	DRSC05954	0
FBgn0031769	CG9135	DRSC03144	0
FBgn0032926	CG9243	DRSC03164	0
FBgn0032923	CG9248	DRSC03167	0
FBgn0032485	CG9426	DRSC03219	0
FBgn0030772	CG9699	DRSC20205	3
FBgn0015372	chrw	DRSC04678	0
FBgn0026084	cib	DRSC18660	0
FBgn0020503	CLIP-190	DRSC03283	3
FBgn0013765	cnn	DRSC07596	0
FBgn0000340	cno	DRSC12374	814
FBgn0000283	Cp190	DRSC16607	0
FBgn0011570	cpb	DRSC00809	1
FBgn0025864	Crag	DRSC18454	0
FBgn0028869	CSN1a	DRSC01951	0
FBgn0000384	cta	DRSC03769	0
FBgn0011202	dia	DRSC03519	9
FBgn0033739	Dyb	DRSC07208	1
FBgn0000556	Ef1[alpha]48D	DRSC07421	1
FBgn0034627	EfSec	DRSC04571	0
FBgn0000578	ena	DRSC07610	10
FBgn0028380	fal	DRSC00806	2
FBgn0024238	Fim	DRSC20243	0
FBgn0035739	form3	DRSC10297	2
FBgn0004435	G[alpha]49B	DRSC07432	3
FBgn0001105	G[beta]13F	DRSC20247	0
FBgn0004623	G[beta]76C	DRSC11174	0
FBgn0004921	G[gamma]1	DRSC07435	0
FBgn0028433	G[gamma]30A	DRSC02715	0
FBgn0033714	garz	DRSC07193	0
FBgn0021873	Gef26	DRSC03231	0

FBgn0035574	Gef64C	DRSC08318	1
FBgn0035574	Gef64C	DRSC08319	10
FBgn0035574	Gef64C	DRSC36490	0
FBgn0010225	Gel	DRSC12350	1
FBgn0001104	G-i[alpha]65A	DRSC11168	1
FBgn0001122	G-o[alpha]47A	DRSC07430	1
FBgn0030685	Graf	DRSC20131	0
FBgn0026431	Grip75	DRSC03337	0
FBgn0026430	Grip84	DRSC20248	1
FBgn0032960	Grp1	DRSC03694	0
FBgn0001123	G-s[alpha]60A	DRSC04616	0
FBgn0036545	GXIVsPLA2	DRSC10243	1
FBgn0028371	jbug	DRSC11087	0
FBgn0001301	kel	DRSC03554	0
FBgn0001612	l(1)dd4	DRSC20249	0
FBgn0020278	loco	DRSC16989	1
FBgn0002567	ltd	DRSC07522	3
FBgn0029688	lva	DRSC18403	0
FBgn0002645	Map205	DRSC16732	1
FBgn0010342	Map60	DRSC07464	0
FBgn0034282	Mapmodulin	DRSC06956	0
FBgn0029870	Marf	DRSC18329	1
FBgn0015513	mbc	DRSC16995	0
FBgn0015513	mbc	DRSC36492	0
FBgn0013951	Mer	DRSC20259	0
FBgn0036333	MICAL	DRSC09829	0
FBgn0021776	mira	DRSC16998	4
FBgn0031885	Mnn1	DRSC03368	0
FBgn0011661	Moe	DRSC18684	0
FBgn0002789	Mp20	DRSC07474	2
FBgn0010070	Msp-300	DRSC03370	0
FBgn0027948	msps	DRSC17004	0
FBgn0039532	Mtl	DRSC16751	0
FBgn0015269	Nf1	DRSC16758	2
FBgn0004625	norpA	DRSC18806	0
FBgn0031437	p16-ARC	DRSC00730	0
FBgn0026193	par-1	DRSC07660	0
FBgn0029137	Patsas	DRSC02975	0
FBgn0041789	Pax	DRSC02651	0
FBgn0041789	Pax	DRSC02652	0
FBgn0003041	pbl	DRSC26301	0
FBgn0003041	pbl	DRSC33335	0

FBgn0003041	pbl	DRSC33336	0
FBgn0003041	pbl	DRSC11381	3
FBgn0024314	Plap	DRSC00678	0
FBgn0033075	Pld	DRSC04854	1
FBgn0013726	pnut	DRSC07666	2
FBgn0029903	pod1	DRSC18362	78
FBgn0014009	Rab2	DRSC05017	1
FBgn0037072	Rab26	DRSC11837	58
FBgn0005586	Rab3	DRSC07523	0
FBgn0031882	Rab30	DRSC03137	4
FBgn0031090	Rab35	DRSC20691	1
FBgn0027505	rab3-GAP	DRSC02001	0
FBgn0014010	Rab5	DRSC00777	2
FBgn0015797	Rab6	DRSC03404	2
FBgn0015795	Rab7	DRSC16810	0
FBgn0032782	Rab9	DRSC03281	0
FBgn0015794	Rab-RP4	DRSC18702	1
FBgn0030200	RabX2	DRSC18234	7
FBgn0051118	RabX4	DRSC22189	1
FBgn0010333	Rac1	DRSC08688	2
FBgn0033881	RacGAP50C	DRSC07575	0
FBgn0033881	RacGAP50C	DRSC33345	0
FBgn0003346	RanGap	DRSC22003	1
FBgn0036497	ran-like	DRSC10918	0
FBgn0025806	Rap2l	DRSC04646	0
FBgn0014015	Rapgap1	DRSC03406	0
FBgn0027375	RecQ5	DRSC11266	0
FBgn0041191	Rheb	DRSC12148	0
FBgn0014020	Rho1	DRSC07530	0
FBgn0036980	RhoBTB	DRSC11877	0
FBgn0039883	RhoGAP100F	DRSC15409	0
FBgn0039898	RhoGAP102A	DRSC17145	0
FBgn0030808	RhoGAP15B	DRSC19924	1
FBgn0030893	RhoGAP16F	DRSC20025	0
FBgn0030986	RhoGAP18B	DRSC20047	0
FBgn0031118	RhoGAP19D	DRSC20499	1
FBgn0025836	RhoGAP1A	DRSC20649	0
FBgn0034249	RhoGAP54D	DRSC06990	1
FBgn0029778	RhoGAP5A	DRSC18294	4
FBgn0036257	RhoGAP68F	DRSC10717	0
FBgn0036518	RhoGAP71E	DRSC10232	1
FBgn0038747	RhoGAP92B	DRSC15647	0

FBgn0038853	RhoGAP93B	DRSC15488	0
FBgn0026375	RhoGAPp190	DRSC20099	2
FBgn0023172	RhoGEF2	DRSC07531	1
FBgn0023172	RhoGEF2	DRSC29373	0
FBgn0035128	RhoGEF3	DRSC08266	0
FBgn0035761	RhoGEF4	DRSC11011	1
FBgn0014380	RhoL	DRSC16824	1
FBgn0015803	rtGEF	DRSC22207	0
FBgn0038947	sar1	DRSC17049	0
FBgn0038947	sar1	DRSC22115	11
FBgn0041781	SCAR	DRSC03426	0
FBgn0037357	sec23	DRSC12387	0
FBgn0014029	Septin-2	DRSC16855	0
FBgn0026361	Septin-5	DRSC06843	1
FBgn0003392	shi	DRSC20373	2
FBgn0013733	shot	DRSC25070	0
FBgn0019652	SIF	DRSC22828	2
FBgn0001961	Sop2	DRSC03438	1
FBgn0001965	Sos	DRSC03439	5
FBgn0011716	spdo	DRSC17062	1
FBgn0003520	stau	DRSC07698	0
FBgn0024277	trio	DRSC08527	78
FBgn0011726	tsr	DRSC04718	0
FBgn0040068	vav	DRSC24485	0
FBgn0022960	vimar	DRSC05026	0
FBgn0004049	yrt	DRSC16559	0

References

1. J. Lindblad, C. Wahlby, E. Bengtsson, A. Zaltsman, *Cytometry A* **57**, 22 (Jan, 2004).
2. A. J. Ridley, A. Hall, *Cell* **70**, 389 (Aug 7, 1992).
3. A. J. Ridley, H. F. Paterson, C. L. Johnston, D. Diekmann, A. Hall, *Cell* **70**, 401 (Aug 7, 1992).
4. R. Lin, S. Bagrodia, R. Cerione, D. Manor, *Curr Biol* **7**, 794 (Oct 1, 1997).
5. R. Lin, R. A. Cerione, D. Manor, *J Biol Chem* **274**, 23633 (Aug 13, 1999).
6. N. Fidyk, J. B. Wang, R. A. Cerione, *Biochemistry* **45**, 7750 (Jun 27, 2006).
7. K. L. Rossman, C. J. Der, J. Sondek, *Nat Rev Mol Cell Biol* **6**, 167 (Feb, 2005).
8. A. A. Kiger *et al.*, *J Biol* **2**, 27 (2003).
9. K. Murayama *et al.*, *J Biol Chem* (Dec 26, 2006).
10. M. Sone *et al.*, *Science* **275**, 543 (Jan 24, 1997).
11. R. Marone *et al.*, *Nat Cell Biol* **6**, 515 (Jun, 2004).

12. I. Nishimura, Y. Yang, B. Lu, *Cell* **116**, 671 (Mar 5, 2004).
13. A. F. Baas *et al.*, *Embo J* **22**, 3062 (Jun 16, 2003).
14. G. J. Bashaw, H. Hu, C. D. Nobes, C. S. Goodman, *J Cell Biol* **155**, 1117 (Dec 24, 2001).
15. T. Schwabe, R. J. Bainton, R. D. Fetter, U. Heberlein, U. Gaul, *Cell* **123**, 133 (Oct 7, 2005).
16. H. F. Paterson *et al.*, *J Cell Biol* **111**, 1001 (Sep, 1990).
17. D. A. Harrison, R. Binari, T. S. Nahreini, M. Gilman, N. Perrimon, *Embo J* **14**, 2857 (Jun 15, 1995).
18. K. R. Castleman, *Digital Image Processing* (Prentice Hall, Upper Saddle River, NJ, 1996), pp. 667.
19. K. V. Mardia, *Biometrika* **57**, 519 (Dec., 1970, 1970).
20. P. Pudil, J. Novovicova, J. Kittler, *Pattern Recognition Letters* **15**, 1119 (1994).
21. M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein, *Proc Natl Acad Sci U S A* **95**, 14863 (Dec 8, 1998).
22. M. Ashburner *et al.*, *Nat Genet* **25**, 25 (May, 2000).
23. S. Parks, E. Wieschaus, *Cell* **64**, 447 (Jan 25, 1991).
24. K. K. Nikolaidou, K. Barrett, *Curr Biol* **14**, 1822 (Oct 26, 2004).
25. S. L. Rogers, U. Wiedemann, U. Hacker, C. Turck, R. D. Vale, *Curr Biol* **14**, 1827 (Oct 26, 2004).
26. R. T. Premont *et al.*, *Proc Natl Acad Sci U S A* **95**, 14082 (Nov 24, 1998).
27. E. E. Sander, J. P. ten Klooster, S. van Delft, R. A. van der Kammen, J. G. Collard, *J Cell Biol* **147**, 1009 (Nov 29, 1999).
28. J. P. Ten Klooster *et al.*, *Biochem J* **397**, 39 (Jul 1, 2006).
29. M. Padash Barmchi, S. Rogers, U. Hacker, *J Cell Biol* **168**, 575 (Feb 14, 2005).
30. U. Hacker, N. Perrimon, *Genes Dev* **12**, 274 (Jan 15, 1998).
31. J. Grosshans *et al.*, *Development* **132**, 1009 (Mar, 2005).
32. K. Barrett, M. Leptin, J. Settleman, *Cell* **91**, 905 (Dec 26, 1997).