Supplemental Materials

Molecular Biology of the Cell

Tie et al.

Supplemental materials

Figure S1. The effect of polynomial fitting order on the chromatic shift correction. The mean magnitudes of green and blue vectors (Figure 2C) are shown as green and blue columns respectively. The values of the 4th order columns are out of the display range and are indicated at the top. Error bars, SDs. n=566 beads were analyzed.

Figure S2. The effect of background subtraction values and GalT-mCherry over-expression levels on the LQ. (A) The LQ of GS15 was not significantly affected by the variation of background subtraction values. The background subtraction values were normalized. 46 analyzable Golgi mini-stacks were selected at the background subtraction value of 100 and were color and shape coded in the plot. When the subtraction value successively increased or decreased by 10, some analyzable mini-stacks became non-analyzable as they no longer fulfilled the three selection criteria. LQs of the remaining analyzable ones were calculated, plotted and connected by lines of the same color. The horizontal dotted blue line indicates the LQ (mean ± SEM) of n=46 mini-stacks at the subtraction value of 100. (B,C) LQs of GS15 and TGN46 were not significantly affected by the expression level of GalT-mCherry. For each mini-stack, the LQ was plotted against the mini-stack's normalized total intensity of GalT-mCherry as a red dot. Black lines were used to connect red dots from low to high total intensity of GalT-mCherry. The linear regression fitting line (blue) together with its formula and adjusted R^2 (adj. R^2) is shown. The mean ± SEM of combined LQs is also indicated in each plot. n, the number of Golgi mini-stacks.

Figure S3. Example images of Golgi proteins and RUSH system reporters used in this study. (A) All Golgi proteins (green) used in this study together with GM130 (blue)

and GalT-mCherry (red) under nocodazole treatment. For endogenous GM130 staining, (m) and (r) indicate mouse and rabbit antibody respectively. (B) Example images of RUSH system secretory membrane reporters during our intra-Golgi transport assay. Cells co-expressing GalT-mCherry, reporters and corresponding ER hooks were treated with nocodazole. Biotin was added to release reporters into the secretory pathway. Example images at 0 and 40 min of biotin chase are shown. At 0 min, all reporters display typical ER localization. GM130 (blue) is from endogenous staining. Scale bars, 10 µm.

Figure S4. LQs in different cell lines and the axial length of the Golgi mini-stack. (A) LQs of GFP-Golgin84 and TGN46 in various mammalian cell lines: HeLa (human cervical epithelium), RPE1 (hTERT RPE1, human retinal pigmented epithelium), BSC1 (monkey kidney epithelium), C2C12 (mouse myoblast) and NRK fibroblast (normal rat kidney fibroblast) cells. GFP-Golgin84 was introduced by transient expression while TGN46 was endogenous. Note that TGN46 antibody is human specific and does not work for C2C12 and NRK cells. Error bars, SEMs. $n \geq 45$ Golgi mini-stacks were analyzed for each LQ. (B,C) Histogram of axial lengths of Golgi stacks (distances from GM130 to GalT-mCherry) or Golgi complexes (distances from GM130 to TGN46). The calculated lengths via GLIM were mean 2D projected values. To estimate the length in 3D, the mean 2D projected values were multiplied by $\pi/2$ (see Materials and methods, Projection of unit-length 3D line segment). In B and C, mean ± SEM is also indicated in the plot. n, the number of Golgi mini-stacks.

Figure S5. The tyrosine or acidic cluster motif of furin is sufficient for the endocytic targeting to the TGN. The reporter ss-SBP-GFP-CD8a-furin wild type, Y mutation, AC

mutation or Y+AC mutation was co-expressed with the ER hook ss-Strep-KDEL. To study the endocytic TGN targeting, cells were incubated in the presence of biotin and CD8a mouse monoclonal antibody for 12 hours and processed for immunofluorescence labeling. Wild type, Y mutation and AC mutation localized to the Golgi and were able to target the internalized CD8a antibody to the Golgi; however, Y+AC mutation neither localized to the Golgi nor was it able to target CD8a antibody to the Golgi. Note that cells were not treated with nocodazole. Scale bar, 10 µm.

Figure S6 Wild type furin chimera did not synchronously arrive on the PM during the chase. The reporter ss-SBP-GFP-CD8a-furin wild type or Y+AC mutation was co-expressed with the ER hook ss-Strep-KDEL. Cells were chased by biotin treatment under the same condition as experiments in Figure 7 for various length of time and subsequently incubated with CD8a mouse monoclonal antibody on ice for 1 hour. The surface bound CD8a was revealed by immunofluorescence. Scale bar, 10 µm.

Table S1. LQs of GFP-Golgin84, GS15 and TGN46 from three independently conducted experiments.

Golgi protein	**experiment #1**			**experiment #2**			**experiment #3**		
	mean	SEM	n	mean	SEM	n	mean	SEM	n
GFP-Golgin84	0.29	0.02	138	0.3	0.03	143	0.26[a]	0.03	108
GS15	0.82[a]	0.03	97	0.82	0.02	177	0.83	0.03	150
TGN46	1.47[a]	0.02	203	1.43	0.06	85	1.48	0.05	107

[a]Data is also used in Table 1.

Table S2. LQs obtained by the spinning disk confocal and wide-field microscope.

Golgi protein	**spinning disk confocal**			**wide-field**		
	mean	SEM	n	mean	SEM	n
KDELR	-0.11	0.03	130	-0.14	0.03	163
Golgin97	1.34	0.03	177	1.38	0.04	138
Golgin245	1.39	0.06	127	1.43	0.04	136

Endogenous KDELR, Golgin97 and Golgin245 were immuno-stained. The same glass coverslips were imaged by the confocal and wide-field microscope and the resulting LQs are listed.

Video S1 Three-color live cell imaging of cells expressing GalT-iRFP670, GFP-Golgin-84, and mCherry-GM130. HeLa cells transiently expressing GalT-iRFP670, GFP-Golgin84 and mCherry-GM130 were treated with nocodazole and imaged live for 24 min with a time interval of 30 sec. The time lapse corresponds to Figure 5. The time stamp represents min:sec.

Note 1. **The MATLAB code for chromatic shift correction by polynomial fitting.**

Function:	The software is first calibrated or trained by beads before chromatic shift-correction. The order of the polynomial fitting can be selected.

For calibration:

Input:	1) Raw coordinates of centers of beads are saved as a CSV file. The coordinates must be arranged as: Rx, Ry, Gx, Gy, Bx and By. R=red; G=green; B=far red. 2) The order of the polynomial fitting.
Output:	A model file containing the calibration information is generated. It is used for shift correction.

For shift correction:

Input:	1) The model file generated in the calibration. 2) Raw coordinates of centers of Golgi mini-stacks are saved as a CSV file. The coordinates must be arranged as: Rx, Ry, Gx, Gy, Bx and By. R=red; G=green; B=far red. 3) The order of the polynomial fitting used in the calibration.
Output:	A CSV file containing shift-corrected coordinates arranged as Rx, Ry, Gx, Gy, Bx and By. Rx and Ry are the same as raw values since the red channel is free of chromatic aberration by definition.

```
%-----------------------
% runme.m
%-----------------------

%MAIN ENTRY of the software
%
%   PARAMETERS TO BE SET
%   train_filename: csv file containing the x,y locations of each of the
%   three colour channels for N samples
%
```

```
%   d: degree of the polynomial to be used for fitting (set to 1 by default)
%
%   val_filename: csv file containing the x,y locations of each of the
%   three colour channels for samples in the validation set
%
%   test_filename: csv file containing the x,y locations of each of the
%   three colour channels for samples in the test set
%

clear all; close all; clc;
mean_err = [];

% set the degree of the polynomial to be used
d = 1;
my_clock = '20120518_1800';
                          % a string that contains the current time stamp, here is an
example
csv_predfname= sprintf('pred_degree_%d_%s.csv', d, my_clock);
% the prediction result is saved in this file

% Train the model to obtain the coefficients of the polynomial
[drgx_coeff, drgy_coeff, dcgx_coeff, dcgy_coeff] = my_train_new('20120518-
BeadsCentroid-1-7-allInOneSheet.csv', d);

% validate the trained model (optional)
mean_err(d+1,:) = my_val_new('20120518-BeadsCentroid-a.csv', drgx_coeff,
drgy_coeff, dcgx_coeff, dcgy_coeff, d);

% Use the trained model on the test data to obtain predictions
[x_r2gfp_res, y_r2gfp_res, x_c2gfp_res, y_c2gfp_res] = my_test_new('20120518-
BeadsCentroid-a.csv', csv_predfname, drgx_coeff, drgy_coeff, dcgx_coeff, dcgy_coeff,
d);



%----------------------
% my_train_new.m
%----------------------

function  [drgx_coeff, drgy_coeff, dcgx_coeff, dcgy_coeff] =
my_train_new(train_filename, d)
%MY_TRAIN_NEW Use training data to calculate the coeffecients of variable
%degree polynomial, which best fits the observations
%
%   INPUTS
%   train_filename: csv file containing the x,y locations of each of the
%   three colour channels for N samples
%   d: degree of the polynomial to be used for fitting
%
%   OUTPUTS
%   drgx_coeff, drgy_coeff: Mx1 vectors containing coefficients for fitting and
subsequent correcting of X and Y data for G channel
%   dcgx_coeff, dcgy_coeff: Mx1 vectors containing coefficients for fitting and
subsequent correcting of X and Y data for CY5 channel
%
%   M is the number of terms in a polynomial of degree d
%
%
%

% default lookup file name in case it wasn't provided as input
if nargin < 1
   train_filename = 'Beads_Centroid-1-9.csv';
end

% Load the dataset
my_dataset  = my_load_dataset_csv(train_filename);
[dummy, c] = size(my_dataset);
assert(dummy==1);
result_dataset = cell(1, c);

% Extract out the x,y coordinates for the 3 color channels
% Each is a Nx1 vector
[x_gfp, y_gfp, x_rfp, y_rfp, x_cy5, y_cy5] = my_create_vector(my_dataset);

% Calculating the emperically observed shift
delta_x_rfp_gfp = x_gfp - x_rfp;
delta_y_rfp_gfp = y_gfp - y_rfp;

% Building the polynomial and the data matrix A
% Example, for d=1:  a00 * x^0*y^0+ a01 * x^0*y^1+ a10 * x^1*y^0
% and A = [(x_rfp.^0).*(y_rfp.^0) (x_rfp.^0).*(y_rfp.^1) (x_rfp.^1).*(y_rfp.^0)];
[~, A] = make_polynomial(d,x_rfp,y_rfp);
% QR decomposing A
[Q,R]=qr(A,0);

% Treating like a system of linear equations Ax = B
% where x is the coefficients of the polynomial and B = Q'*delta
drgx_coeff = R\(Q'*delta_x_rfp_gfp);
drgy_coeff = R\(Q'*delta_y_rfp_gfp);

% A similar procedure follows for the CY5 correction
delta_x_cy5_gfp = x_gfp - x_cy5;
```

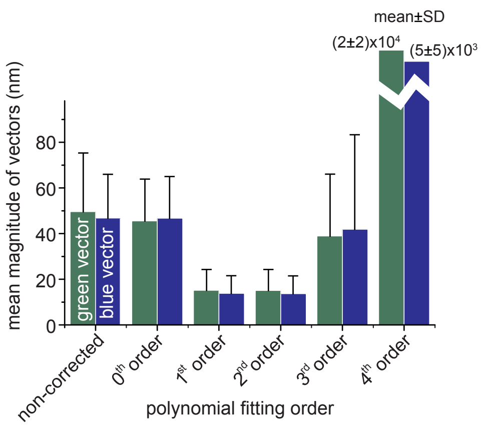
```
delta_y_cy5_gfp = y_gfp - y_cy5;

[~, A] = make_polynomial(d,x_cy5,y_cy5);
[Q,R]=qr(A,0);

dcgx_coeff = R\(Q'*delta_x_cy5_gfp);
dcgy_coeff = R\(Q'*delta_y_cy5_gfp);

% Storing the results and writing them out to a .csv file
result_dataset{1} = [drgx_coeff, drgy_coeff, dcgx_coeff, dcgy_coeff];
csvwrite(['train_degree' '_' int2str(d) '_coeff.csv'], result_dataset);

%----------------------
% my_val_new.m
%----------------------

function  [mean_err] = my_val_new(val_filename, drgx_coeff, drgy_coeff, dcgx_coeff,
dcgy_coeff, d)
%MY_VAL_NEW Validate the trained model on a test set.
%
%   INPUTS
%   val_filename: csv file containing the x,y locations of each of the
%   three colour channels for N samples of the validation data
%   d: degree of the polynomial to be used for fitting
%   drgx_coeff, drgy_coeff, dcgx_coeff, dcgy_coeff: learned coefficients
%
%   OUTPUTS
%   mean_err: The mean error in estimation of the corrected locations. This
%   is a 1x4 vector, containing errors in x,y for the two channels
%

% default lookup file name in case it wasn't provided as input
if nargin < 1
   val_filename = 'Beads_Centroid-10-11.csv';
end

% Load the dataset
my_dataset  = my_load_dataset_csv(val_filename);
[dummy, c] = size(my_dataset);
assert(dummy==1);
mean_err = zeros(1,4);

% Extract out the x,y coordinates for the 3 color channels
% Each is a Nx1 vector
[x_gfp,y_gfp,x_rfp,y_rfp,x_cy5,y_cy5] = my_create_vector(my_dataset);
```

```
% Make the data matrix and multiply with the learned coefficients to get
% the deltas
[~, A] = make_polynomial(d,x_rfp,y_rfp);
delta_x_rfp_gfp_res = A*drgx_coeff;
delta_y_rfp_gfp_res = A*drgy_coeff;

% Add obs. value back to get the predicted location
x_r2gfp_res = delta_x_rfp_gfp_res + x_rfp;
y_r2gfp_res = delta_y_rfp_gfp_res + y_rfp;

% Difference between the actual and the predicted locations
x_tmp_r2g = x_gfp-x_r2gfp_res;
y_tmp_r2g = y_gfp-y_r2gfp_res;

% Calculating the mean errors
mean_err(1) = mean(abs(x_tmp_r2g));
mean_err(2) = mean(abs(y_tmp_r2g));

figure,
subplot(1, 2, 1), scatter(x_tmp_r2g, y_tmp_r2g);
msg = sprintf('d=%d, rfp2gfp', d); title(msg);

% A similar procedure follows for the CY5 correction
[~, A] = make_polynomial(d,x_cy5,y_cy5);

delta_x_cy5_gfp_res = A*dcgx_coeff;
delta_y_cy5_gfp_res = A*dcgy_coeff;

x_c2gfp_res = delta_x_cy5_gfp_res + x_cy5;
y_c2gfp_res = delta_y_cy5_gfp_res + y_cy5;

x_tmp_c2g = x_gfp-x_c2gfp_res;
y_tmp_c2g = y_gfp-y_c2gfp_res;

mean_err(3) = mean(abs(x_tmp_c2g));
mean_err(4) = mean(abs(y_tmp_c2g));

subplot(1, 2, 2), scatter(x_tmp_c2g, y_tmp_c2g);
msg = sprintf('d=%d, cy52gfp', d); title(msg);

result_dataset = [x_gfp, y_gfp, x_r2gfp_res, y_r2gfp_res, x_c2gfp_res, y_c2gfp_res];
csvwrite(['val_degree' '_' int2str(d) '_pred.csv'], result_dataset);

% print the errors out
mean_err
```

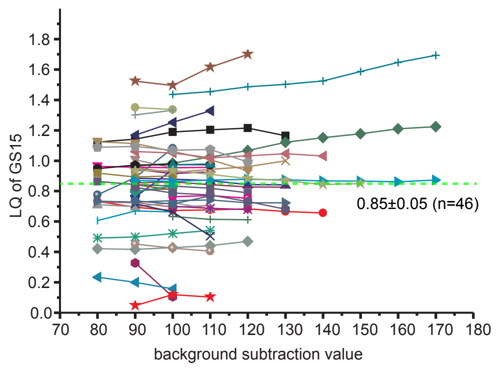
```
%-----------------------
% my_test_new.m
%-----------------------

function  [x_r2gfp_res, y_r2gfp_res, x_c2gfp_res, y_c2gfp_res] =
my_test_new(test_filename, csv_outfname, drgx_coeff, drgy_coeff, dcgx_coeff,
dcgy_coeff, d)
%MY_TEST_NEW Use the learned coeffecients of a polynomial
% to correct for chromatic aberration in the test data
%
%   INPUTS
%   test_filename: csv file containing the x,y locations of the test data
%   d: degree of the polynomial to be used for fitting
%   drgx_coeff, drgy_coeff: Mx1 vectors containing the learned coefficients G channel
correction
%   dcgx_coeff, dcgy_coeff: Mx1 vectors containing the learned coefficients
%   CY5 channel correction
%
%   OUTPUTS
%   x_r2gfp_res, y_r2gfp_res: Corrected coordinates for G channel of size Nx1
%   x_c2gfp_res, y_c2gfp_res: Corrected coordinates for CY5 channel, size Nx1
%
%

% default test file name in case it wasn't provided as input
if nargin < 1
   test_filename = 'Noc-Golgi-Mini-Stack_Centroid.xls';
end

% Load the test data
my_dataset  = my_load_dataset_csv(test_filename);
[~, c] = size(my_dataset);
result_dataset = cell(1, c);

% Extract out the x,y coordinates for the 3 color channels
% Each is a Nx1 vector
[~,~,x_rfp,y_rfp,x_cy5,y_cy5] = my_create_vector(my_dataset);

% GFP correction
% first build the polynomial and the data matrix.
% A*learned coefficients will give delta
[~, A] = make_polynomial(d,x_rfp,y_rfp);
delta_x_rfp_gfp_res = A*drgx_coeff;
delta_y_rfp_gfp_res = A*drgy_coeff;
```

```
% Adding back the original location to delta and obtain the corrected loc.
x_r2gfp_res = delta_x_rfp_gfp_res + x_rfp;
y_r2gfp_res = delta_y_rfp_gfp_res + y_rfp;


% A similar procedure follows for the CY5 correction
[~, A] = make_polynomial(d,x_cy5,y_cy5);
delta_x_cy5_gfp_res = A*dcgx_coeff;
delta_y_cy5_gfp_res = A*dcgy_coeff;

x_c2gfp_res = delta_x_cy5_gfp_res + x_cy5;
y_c2gfp_res = delta_y_cy5_gfp_res + y_cy5;

% Storing the results and writing them out to a .csv file
result_dataset{1} = [x_r2gfp_res, y_r2gfp_res, x_c2gfp_res, y_c2gfp_res];
%csvwrite(['test_degree' '_' int2str(d) '_pred.csv'], result_dataset);
csvwrite(csv_outfname, result_dataset);

%----------------------
% my_create_vector.m
%----------------------

function [x_gfp,y_gfp,x_rfp,y_rfp,x_cy5,y_cy5] = my_create_vector(dataset)
%MY_CREATE_VECTOR Extract location data as separate vectors from the loaded
%cell array
%
%   INPUTS
%   dataset: csv file loaded into memory
%
%   OUTPUTS
%   x_gfp, y_gfp: x,y info for the red channel
%   x_rfp, y_rfp: x,y info for the green channel
%   x_cy5, y_cy5: x,y info for the CY5 channel
%   Example usage:
%   data = my_load_dataset_csv('traindata.csv')
%   [x_gfp,y_gfp,x_rfp,y_rfp,x_cy5,y_cy5] = my_create_vector(data)
%
%

[~,c] = size(dataset);
x_gfp = [];
y_gfp = [];
x_rfp = [];
y_rfp = [];
x_cy5 = [];
y_cy5 = [];
```

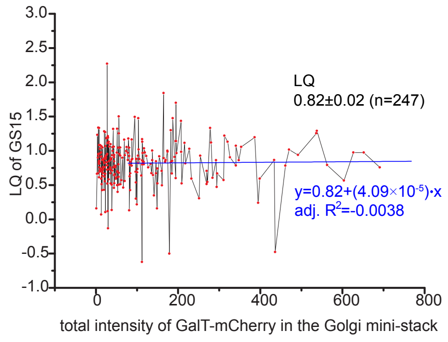
```
for i = 1:c
        vect  = dataset{i};

        x_gfp = [x_gfp; vect(:,1)];
        y_gfp = [y_gfp; vect(:,2)];

        x_rfp = [x_rfp; vect(:,3)];
        y_rfp = [y_rfp; vect(:,4)];

        x_cy5 = [x_cy5; vect(:,5)];
        y_cy5 = [y_cy5; vect(:,6)];
end


%----------------------
% my_load_dataset_csv.m
%----------------------

function dataset = my_load_dataset_csv(filename)
%MY_LOAD_DATASET_CSV Read from a given csv file into a cell array for
%subsequent calculations
%
%   INPUTS
%   filename: csv file containing the x,y locations of each of the
%   three colour channels for N samples
%
%   OUTPUTS
%   dataset: cell array containing the data loaded into memory
%
%

valid_vect = csvread(filename);
dataset{1} = valid_vect ;

%----------------------
% make_polynomial.m
%----------------------

function [function_expr, A] = make_polynomial(degree,x,y)
%MAKE_POLYNOMIAL Given the degree of the required polynomial and the data
%points (bi variate), this function builds the polynomial structure and the
%data matrix
%
%   INPUTS
%   degree: degree of the polynomial to be used for fitting
%   x, y: Nx1 vectors containing the data points
%
%   OUTPUTS
%   function_expr: The polynomial expression (String)
%   A: NxM data matrix where,
%   M is the number of terms in a polynomial of degree d
%   and Aij is the jth polynomial term (without the coefficient) of the ith
%   sample.
%
%

expr = [];
A = [];

for i = 0: degree
   for j=0:degree - i
         % adding the corresponding term to the existing polynomial
         expr = [expr '+a' int2str(i) int2str(j) ' * '  'x^' int2str(i)  ' * ' 'y^' int2str(j) ];
         A = [A (x.^i).*(y.^j)];
   end
end

% removing the , upfront
expr = expr(1, 2:size(expr,2));
function_expr = expr ;

end
```


mean±SD
(2±2)x10^4
(5±5)x10^3
mean magnitude of vectors (nm)
0
20
40
60
80
green vector
blue vector
non-corrected
0th order
1st order
2nd order
3rd order
4th order
polynomial fitting order

A


1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
LQ of GS15
70
80
90
100
110
120
130
140
150
160
170
180
background subtraction value
0.85±0.05 (n=46)

B


3.0
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
LQ of GS15
0
200
400
600
800
total intensity of GalT-mCherry in the Golgi mini-stack
LQ
0.82±0.02 (n=247)
y=0.82+(4.09×10⁻⁵)·x
adj. R²=-0.0038

C

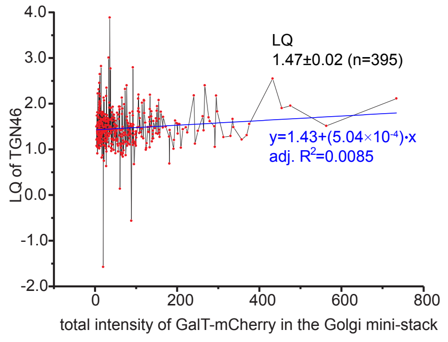

4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
LQ of TGN46
0
200
400
600
800
total intensity of GalT-mCherry in the Golgi mini-stack
LQ
1.47±0.02 (n=395)
y=1.43+(5.04×10⁻⁴)·x
adj. R²=0.0085

Fig. S3-A-1

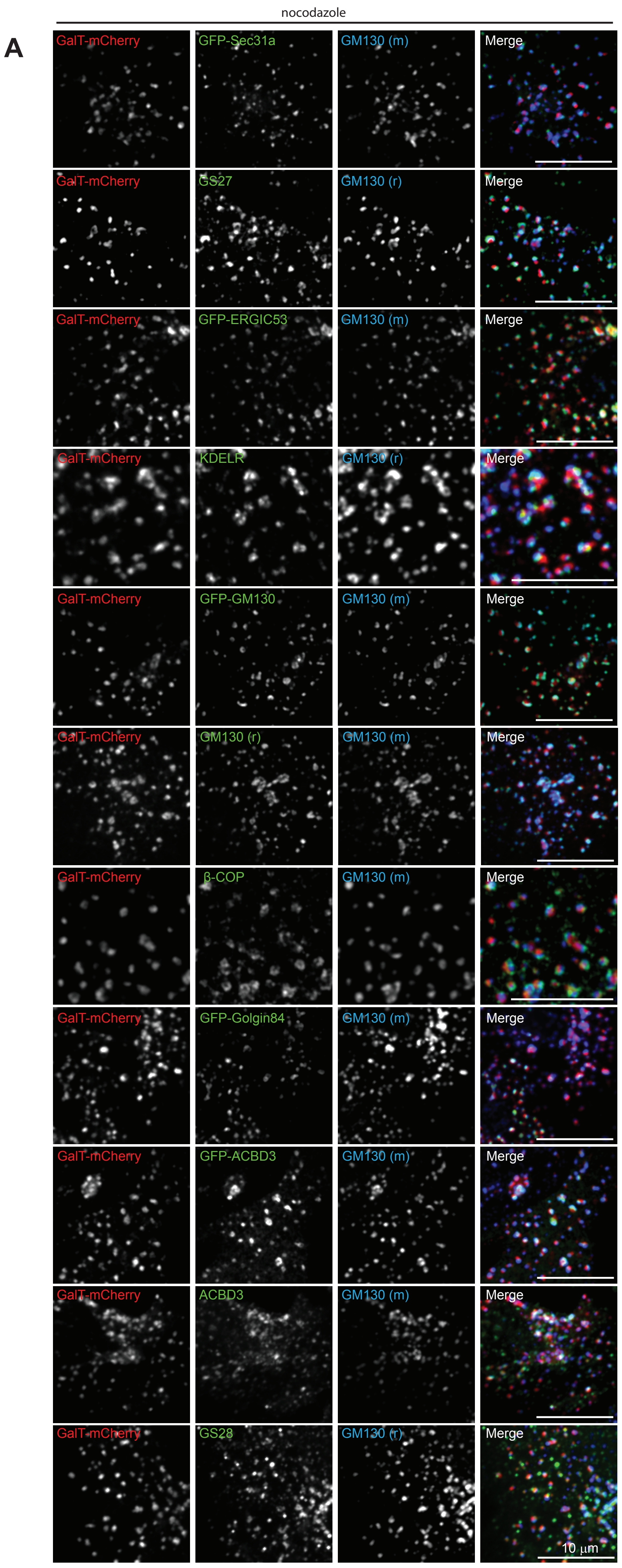

A
nocodazole
GalT-mCherry
GFP-Sec31a
GM130 (m)
Merge
GalT-mCherry
GS27
GM130 (r)
Merge
GalT-mCherry
GFP-ERGIC53
GM130 (m)
Merge
GalT-mCherry
KDELR
GM130 (r)
Merge
GalT-mCherry
GFP-GM130
GM130 (m)
Merge
GalT-mCherry
GM130 (r)
GM130 (m)
Merge
GalT-mCherry
β-COP
GM130 (m)
Merge
GalT-mCherry
GFP-Golgin84
GM130 (m)
Merge
GalT-mCherry
GFP-ACBD3
GM130 (m)
Merge
GalT-mCherry
ACBD3
GM130 (m)
Merge
GalT-mCherry
GS28
GM130 (r)
Merge
10 μm

Fig. S3-A-2

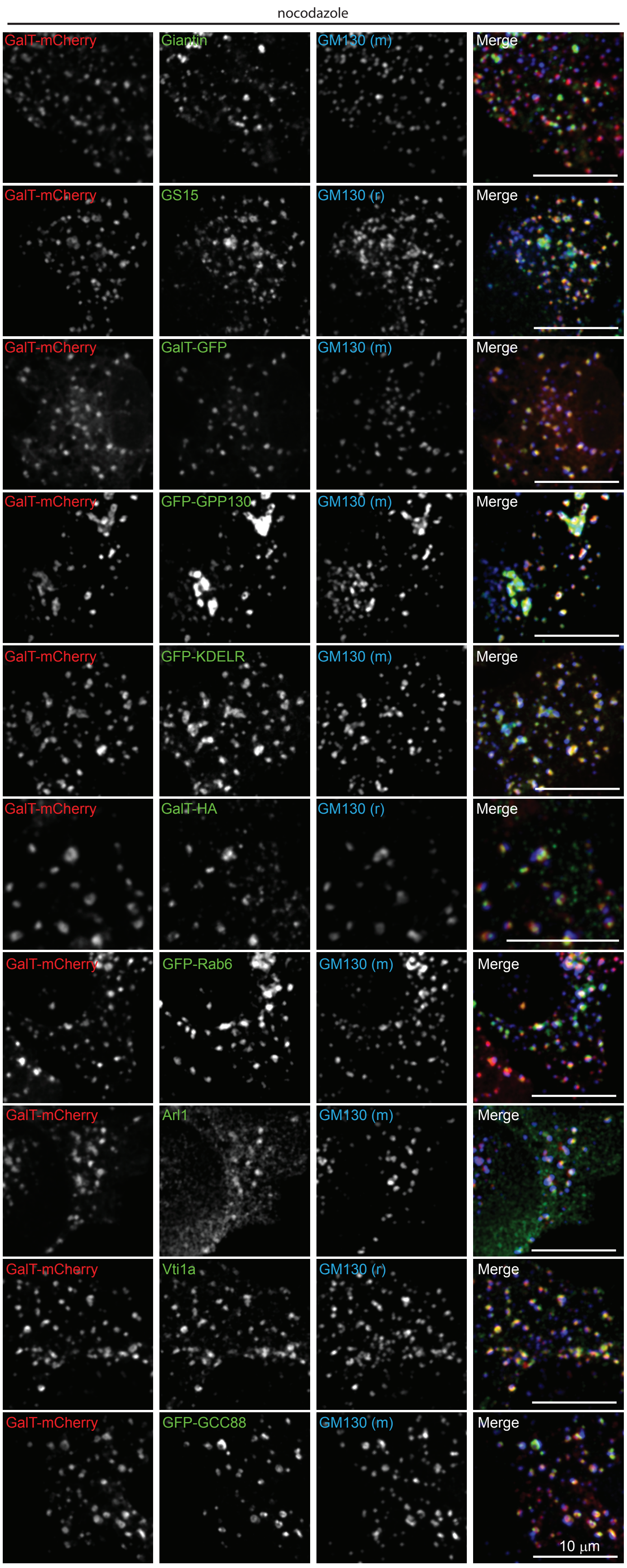
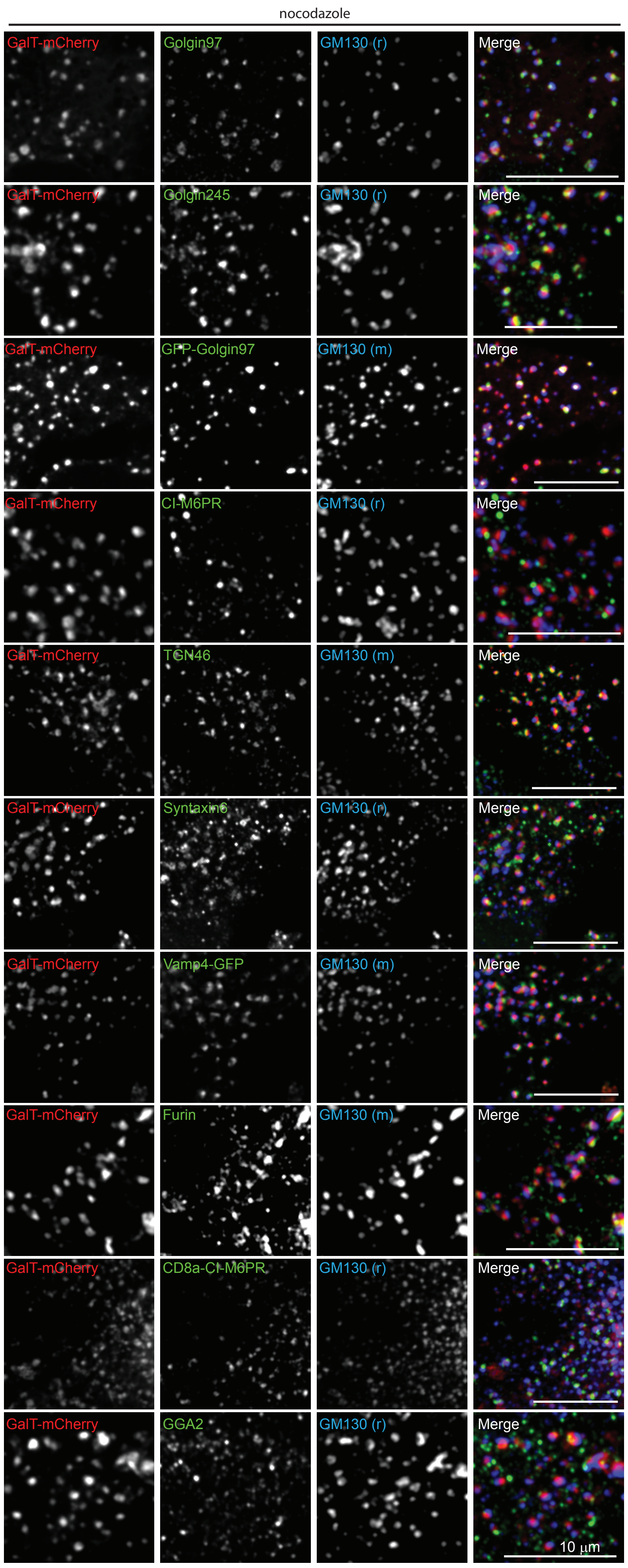
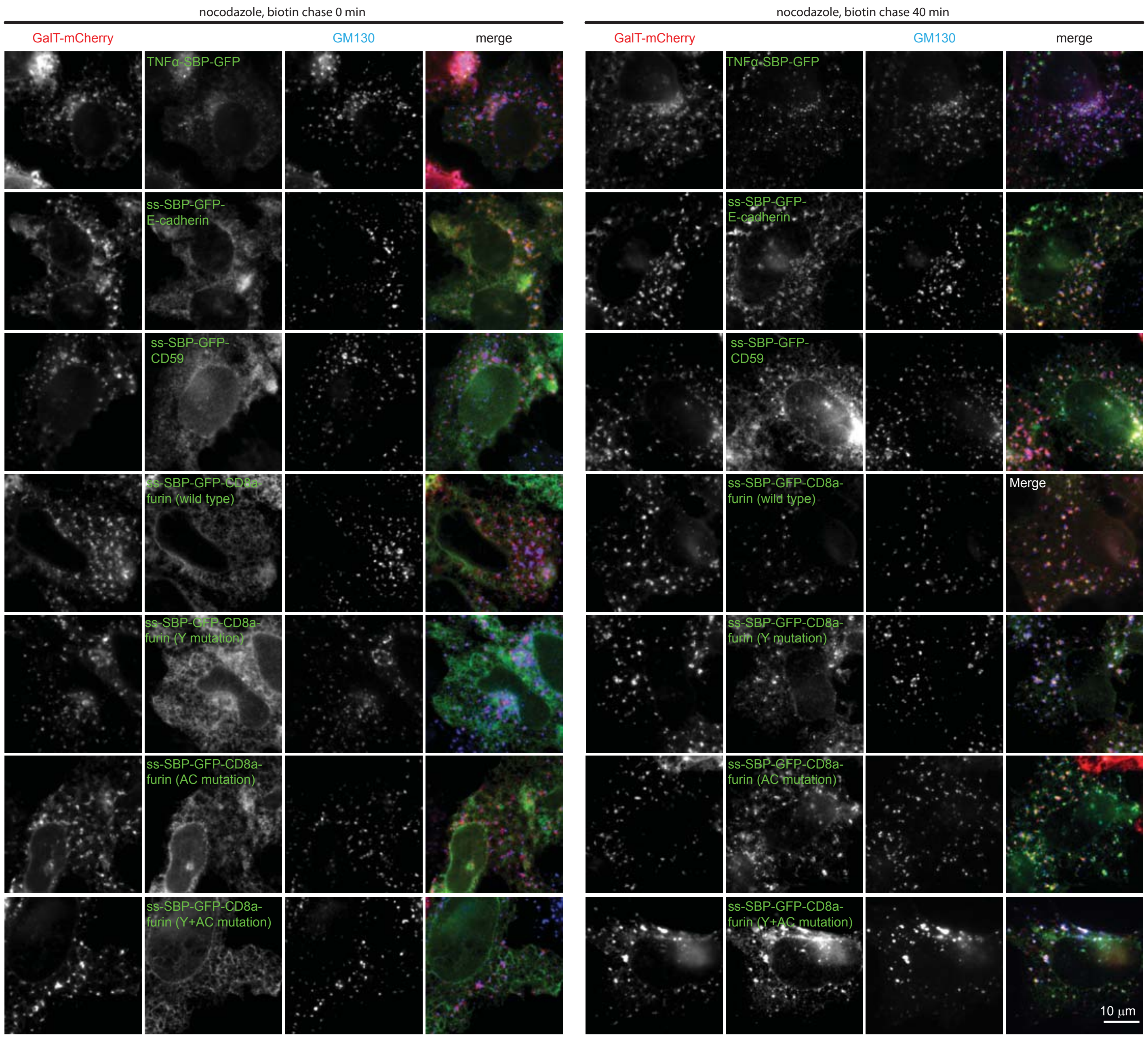

nocodazole
GalT-mCherry
Giantin
GM130 (m)
Merge
GalT-mCherry
GS15
GM130 (r)
Merge
GalT-mCherry
GalT-GFP
GM130 (m)
Merge
GalT-mCherry
GFP-GPP130
GM130 (m)
Merge
GalT-mCherry
GFP-KDELR
GM130 (m)
Merge
GalT-mCherry
GalT-HA
GM130 (r)
Merge
GalT-mCherry
GFP-Rab6
GM130 (m)
Merge
GalT-mCherry
Arl1
GM130 (m)
Merge
GalT-mCherry
Vti1a
GM130 (r)
Merge
GalT-mCherry
GFP-GCC88
GM130 (m)
Merge
10 μm

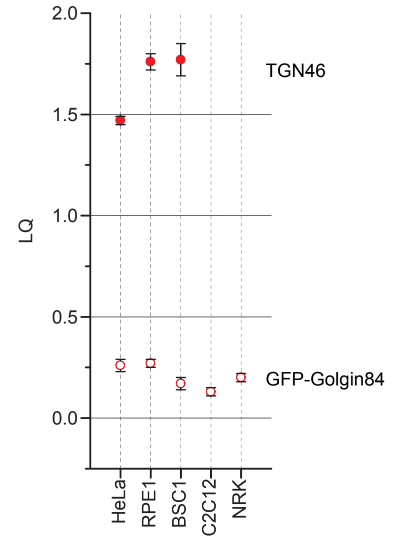
Fig. S3-A-3


nocodazole
GalT-mCherry
Golgin97
GM130 (r)
Merge
GalT-mCherry
Golgin245
GM130 (r)
Merge
GalT-mCherry
GFP-Golgin97
GM130 (m)
Merge
GalT-mCherry
CI-M6PR
GM130 (r)
Merge
GalT-mCherry
TGN46
GM130 (m)
Merge
GalT-mCherry
Syntaxin6
GM130 (r)
Merge
GalT-mCherry
Vamp4-GFP
GM130 (m)
Merge
GalT-mCherry
Furin
GM130 (m)
Merge
GalT-mCherry
CD8a-CI-M6PR
GM130 (r)
Merge
GalT-mCherry
GGA2
GM130 (r)
Merge
10 μm

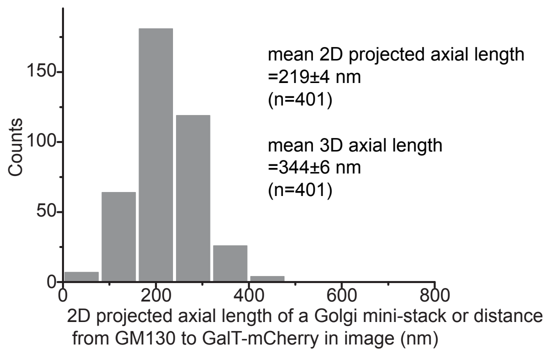
B


nocodazole, biotin chase 0 min
nocodazole, biotin chase 40 min
GalT-mCherry
GM130
merge
TNFα-SBP-GFP
ss-SBP-GFP-E-cadherin
ss-SBP-GFP-CD59
ss-SBP-GFP-CD8a-furin (wild type)
ss-SBP-GFP-CD8a-furin (Y mutation)
ss-SBP-GFP-CD8a-furin (AC mutation)
ss-SBP-GFP-CD8a-furin (Y+AC mutation)
Merge
10 μm

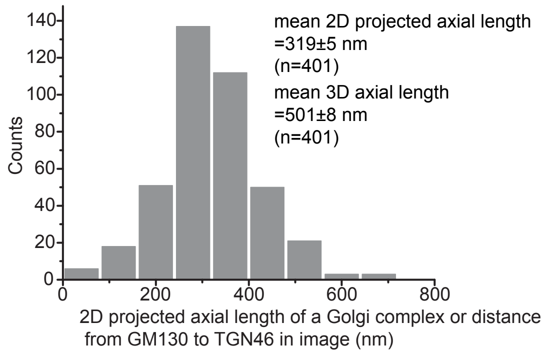
A

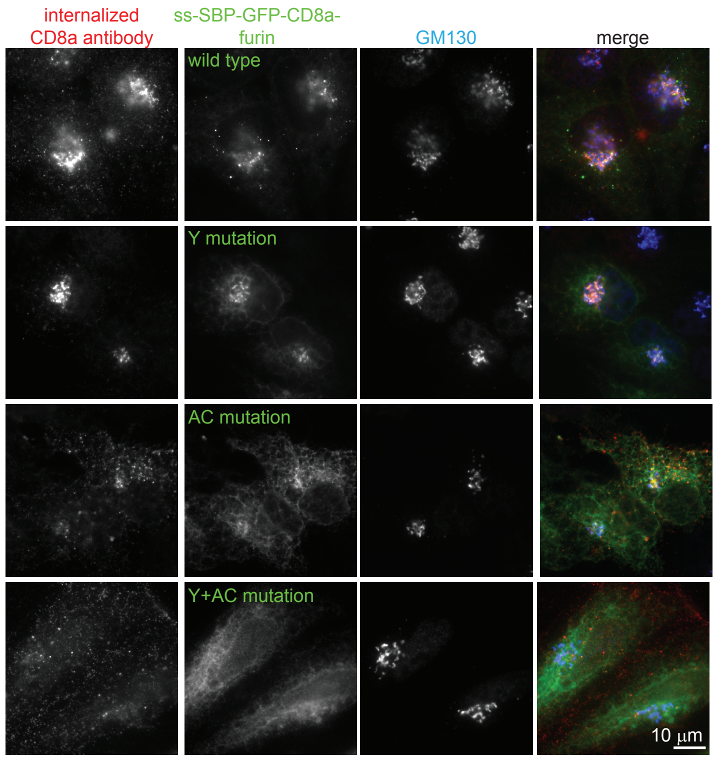

2.0
1.5
1.0
0.5
0.0
LQ
TGN46
GFP-Golgin84
HeLa
RPE1
BSC1
C2C12
NRK

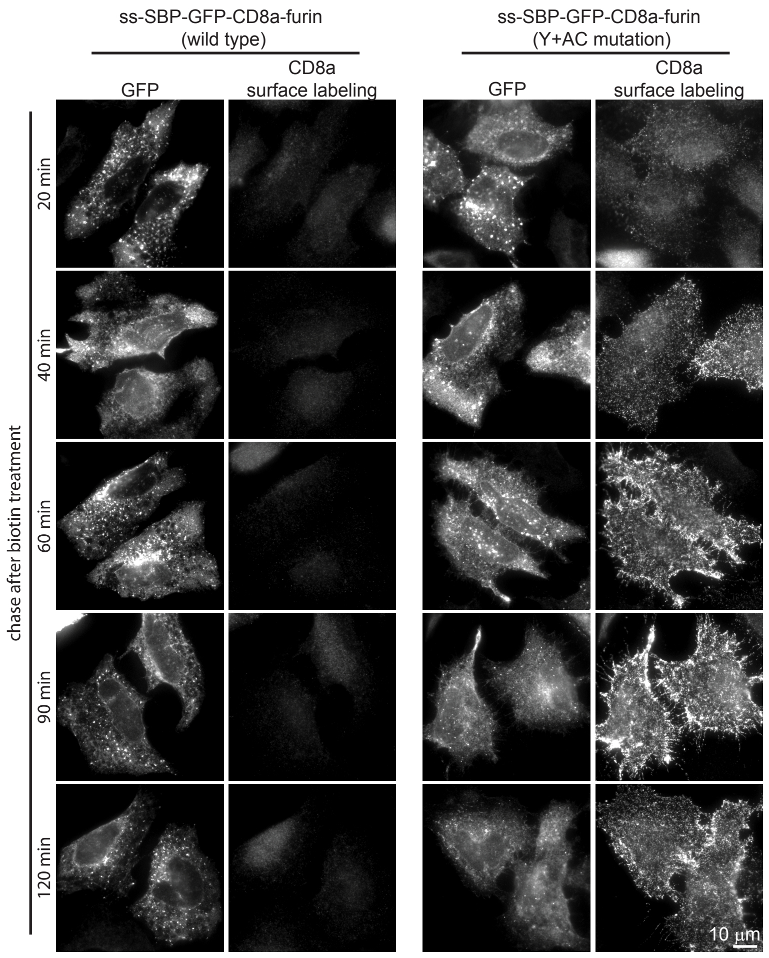
B


150
100
50
0
Counts
mean 2D projected axial length
=219±4 nm
(n=401)
mean 3D axial length
=344±6 nm
(n=401)
0
200
400
600
800
2D projected axial length of a Golgi mini-stack or distance from GM130 to GalT-mCherry in image (nm)

C


140
120
100
80
60
40
20
0
Counts
mean 2D projected axial length
=319±5 nm
(n=401)
mean 3D axial length
=501±8 nm
(n=401)
0
200
400
600
800
2D projected axial length of a Golgi complex or distance from GM130 to TGN46 in image (nm)


internalized CD8a antibody
ss-SBP-GFP-CD8a-furin
GM130
merge
wild type
Y mutation
AC mutation
Y+AC mutation
10 μm


ss-SBP-GFP-CD8a-furin
(wild type)
ss-SBP-GFP-CD8a-furin
(Y+AC mutation)
GFP
CD8a
surface labeling
GFP
CD8a
surface labeling
chase after biotin treatment
20 min
40 min
60 min
90 min
120 min
10 μm