

```
1 # Supplementary File 1- R Script for Figure 1, Figure 1- figure supplement 4
2 #
3 #
4 # This is a program to calculate various parameters for E-cadherin quantifications-----
5 #
6 #
7 #
8 # "rawdata" contains data read from the csv file(provided by user). It should contain the following
9 columns:
10 # 1) Genotype
11 # 2) Cell name
12 # 3) Mean
13 # 4) Length
14 # 5) Boundary -- If you are doing cell wise analysis(put same letter in the whole column in case
15 analysis is cell wise and by boundaries)
16 #
17 #
18 # "stack.all" contains stack wise calculated data which includes:
19 # 1) Height at a particular stack
20 # 2) Normalized Height
21 # 3) Normalized Ecad
22 # 4) Absolute Ecad at each stack
23 #
24 # "cell.all" contains cell wise data. Which includes the following:
25 # 1) The total Height of the Cell
26 # 2) Total Ecad in the cell
27 #
28 # "genotype.all" contains genotype wise data which includes the following:
29 # 1) Mean apical perimeter
30 # 2) Median apical perimeter
31 # 3) Mean Total Ecad per cell
32 # 4) Median Total Ecad per cell
33 # 5) Mean Height of the cell
34 # 6) Median Height of the cell
35 # 7) First quantile of Height of cell
36 # 8) Third quantile of height of cell
37 #
38 # "lofit.model" contains the lofit data and the height at which the levels were predicted.
39 #
40 # "apical.perimeter" contains the apical most stack used for apical perimeter analysis
41 #
42 #
43 #
44 #
45
46 #clears memory-----
47 rm(list = ls())
48
```

```

49 #sets a working directory-----
50 #This is where all your graphs will be saved
51
52 setwd(
53   "C:/Users/prata/Desktop/Test for paper/"
54 )
55
56 #libraries required for the program to work.-----
57 # List of libraries required for the program to run
58 pkg <- c("scales", "locfit", "tidyr", "purrr", "dplyr", "ggplot2", "broom",
59         "data.table", "RCurl", "stringr", "grid", "gridExtra", "RColorBrewer")
60
61 # Check if the libraries are installed and install libraries not present
62 new.pkg <- pkg[!(pkg %in% installed.packages())]
63 if (length(new.pkg)) {
64   install.packages(new.pkg)
65 }
66 rm("pkg", "new.pkg")
67
68 #load libraries in the memory
69 library(scales)
70 library(locfit)
71 library(tidyr)
72 library(purrr)
73 library(dplyr)
74 library(ggplot2)
75 library(data.table)
76 library(RCurl)
77 library(stringr)
78 library(grid)
79 library(gridExtra)
80 library(RColorBrewer)
81
82 #stops warnings from being displayed
83 options(warn = -1)
84
85
86
87 # read raw data-----
88 # the file should should have the following components:
89 # Genotype      Cell name      Area      Mean      Boundary
90
91 #Folder and file names where your input file is saved
92
93 Folderadd <-
94   "C:/Users/prata/Desktop/Test for paper/"
95 filename <-
96   "Figure 1- source data 1"

```

```

97  format <- ".csv"
98  fulladdress <- paste(Folderadd, filename, format, sep = "")
99
100 rawdata <-
101   read.csv(fulladdress,
102           header = TRUE,
103           sep = ",") %>% na.omit()
104
105   # data arranging-----
106
107   #To have Genotype and Cell name, Boundary order as in the file (Otherwise it comes alphabetically
108   which may not be desirable)
109   rawdata$Genotype <-
110     factor(rawdata$Genotype, levels = unique(rawdata$Genotype))
111   rawdata$Cell.name <-
112     factor(rawdata$Cell.name, levels = unique(rawdata$Cell.name))
113   rawdata$Boundary <-
114     factor(rawdata$Boundary, levels = unique(rawdata$Boundary))
115
116
117   #group by genotype and cell name calculate absolute ecad for each stack, and nest the data to to
118   help calculations on groups
119   by_cellname <- rawdata %>%
120     group_by(Genotype, Cell.name) %>%
121     mutate(AbsoluteEcad = Mean * Area) %>%
122     nest()
123
124
125   #function to find height
126   rawheight <- function(df) {
127     height <- vector("double", nrow(df))
128     for (i in seq_along(df$Mean)) {
129       height[i] <- (i - 1) * 0.28 #the height of the z stack
130     }
131     height
132   }
133
134
135   #function to normalize height
136   normheightfun <- function(df) {
137     nheight <- vector("double", nrow(df))
138     maxh <- nrow(df)
139     minh <- 1
140     for (i in seq_along(df$Mean)) {
141       nheight[i] <- ((i) - minh) / (maxh - minh)
142     }
143     nheight
144   }

```

```

145
146 #function to normalize ecad levels
147 normecadfun <- function(df) {
148   necad <- vector("double", nrow(df))
149   maxe <- max(df$Mean)
150   mine <- min(df$Mean)
151   for (i in seq_along(df$Mean)) {
152     necad[i] <- (df$Mean[i] - mine) * 100 / (maxe - mine)
153   }
154   necad
155 }
156
157 #dataframe which now has height per stack, normalized height, normalized ecad per genotype-cell
158 name
159 stack.all <- by_cellname %>%
160   mutate(
161     Stackheight = data %>% map(rawheight),
162     Normalizedheight = data %>% map(normheightfun),
163     NormalizedEcad = data %>% map(normecadfun),
164     Uniquecellname = paste(Genotype, Cell.name)
165   ) %>% unnest()
166
167
168 #to know the total ecad in a cell and height of a cell
169 cell.all <- stack.all %>%
170   group_by(Genotype, Cell.name) %>%
171   summarize(Total_Ecad = sum(AbsoluteEcad),
172     Height_of_cell = max(Stackheight))
173
174 #to fit linear regression to find the abnormal polarized distribution-----
175
176 #function for linear regression
177
178 #loffit -----
179 # loffit will fit the data cell wise and predict values at given cell heights
180
181 #nest the data in stack.all with normalized height to use in loffit for prediction
182 loffit.input <- stack.all %>%
183   group_by(Genotype, Cell.name) %>%
184   nest()
185
186 # loffit.input$data[[1]]
187
188
189 #cell heights chosen to interpolate ecad levels
190 predictvalues <- c(0.1, 0.3, 0.5, 0.7, 0.9)
191
192 #To calculate the y-axis limits in loffit graphs

```

```

193 ylimmax <- max(rawdata$Mean) + 250
194 ylimmin <- min(rawdata$Mean) - 100
195
196
197 #function to fit locfit
198 locfit_model <- function(df) {
199   fit <- locfit(Mean ~ Normalizedheight, data = df)
200   plot(
201     fit,
202     get.data = TRUE,
203     ylim = c(ylimmin, ylimmax),
204     xlim = c(0, 1),
205     xlab = "Normalized Cell Height",
206     ylab = "Ecad Level(in AU)"
207   )
208   par(new = TRUE) #To superimpose graphs
209   predictions <- predict(fit, predictvalues)
210
211 }
212
213 #to plot the locfit graphs
214 #to start a new graph page and close all old ones,
215 # so that new graphs don't get superimposed on the old ones
216 plot.new()
217 dev.off()
218 plot.new()
219
220
221
222
223 #add locfit data to grouped cells and add cell height(on which prediction is made) to each cell
224 locfit.model <- locfit.input %>%
225   mutate(fitvalues = data %>% map(locfit_model)) %>%
226   unnest(fitvalues) %>%
227   group_by(Genotype, Cell.name) %>%
228   mutate(Cell_height = predictvalues)
229 locfit.model <- locfit.model %>% group_by(Genotype, Cell.name)
230
231 #to save graph for locfit data
232 dev.copy(
233   svg,
234   file = paste0(filename, "_locfitgraphs.svg"),
235   width = 4,
236   height = 4
237 )
238
239 dev.off ()
240

```

```

241
242 #for creating groups for each cell for ggplot graph of lines -----
243
244 graph.lines <- stack.all %>% mutate (Uniquecellname = paste(Genotype, Cell.name))
245
246
247 #to find out the apical perimeter of the cell
248
249 apical.perimeter <- graph.lines %>%
250   group_by(Genotype, Cell.name) %>%
251   slice(1) %>% select(Genotype, Cell.name, Length)
252
253 cell.all <- cell.all %>% full_join(apical.perimeter, by = c("Genotype", "Cell.name"))
254
255 cell.all <- rename(cell.all, "Apical Perimeter" = Length)
256
257
258 # to calculate Genotype wise parameters-----
259 genotype.peri <- apical.perimeter %>% group_by(Genotype) %>%
260   summarise(Mean_peri = mean(Length),
261             Median_peri = median(Length))
262
263
264 genotype.summary <- cell.all %>% group_by(Genotype) %>%
265   summarise(
266     Mean_TotalEcad = mean(Total_Ecad),
267     Median_TotalEcad = median(Total_Ecad),
268     Mean_Heightofcell = mean(Height_of_cell),
269     Median_Heightofcell = median(Height_of_cell),
270     First_quantile_Height = quantile(Height_of_cell, prob = 0.25),
271     Third_quantile_Height = quantile(Height_of_cell, prob = 0.75),
272     count = n()
273   )
274
275 #final dataframe for all calculated parameters by merging the above two data frames
276 genotype.all <- genotype.peri %>% full_join(genotype.summary, by = "Genotype")
277
278 #writing csv-----
279
280
281 #write csv with height, normalized height,absolute ecad in each stack
282 write.csv(
283   stack.all,
284   file = paste0(filename, "_stackwise_calculations.csv"),
285   row.names = FALSE
286 )
287
288 #write csv with cell level parameters

```

```

289 write.csv(
290   cell.all,
291   file = paste0(filename, "_cellwise_calculations.csv"),
292   row.names = FALSE
293 )
294
295 #write csv with locfit fitdata
296 write.csv(locfit.model,
297   file = paste0(filename, "_locfitdata.csv"),
298   row.names = FALSE)
299
300 #write csv with genotype level parameters
301 write.csv(
302   genotype.all,
303   file = paste0(filename, "_genotypewise_calculations.csv"),
304   row.names = FALSE
305 )
306
307 #graphs-----
308
309 #to plot boxplots for Total Ecad vs Genotype-----
310 p <-
311 ggplot(data = cell.all, aes(x = Genotype, y = Total_Ecad, fill = Genotype)) +
312 theme(
313   plot.title = element_text(hjust = 0.5),
314   text = element_text(size = 14),
315   legend.position = "none",
316   axis.text.x = element_text(angle = 45, hjust = 1)
317 ) +
318 labs(title = "Total E-cadherin", x = "Genotype", y = "E-cadherin Levels(in AU)") +
319 scale_y_continuous(labels = comma) +
320 theme(
321   plot.title = element_text(hjust = 0.5),
322   legend.position = "none",
323   text = element_text(size = 14)
324 ) +
325 geom_boxplot(outlier.shape = NA, alpha = 0.7) +
326 geom_point(position = position_jitter(width = .1),
327   shape = 21,
328   alpha = 0.4)#+
329 # coord_cartesian(ylim = c(0, 800000))
330 print(p)
331 dev.copy(
332   svg,
333   file = paste0(filename, "_Total Ecad.svg"),
334   width = 4,
335   height = 4
336 )

```

```

337 dev.off ()
338
339
340
341 #to plot boxplots for Height of cell vs Genotype-----
342 q <-
343 q <-
344 ggplot(data = cell.all, aes(x = Genotype, y = Height_of_cell, fill = Genotype)) +
345 theme(
346   plot.title = element_text(hjust = 0.5),
347   text = element_text(size = 14),
348   legend.position = "none",
349   axis.text.x = element_text(angle = 45, hjust = 1)
350 ) +
351 labs(title = "Height of Cells",
352       x = "Genotype",
353       y = expression(paste("Height(in ", mu, "m)"))) +
354 scale_y_continuous(labels = comma) +
355 theme(
356   plot.title = element_text(hjust = 0.5),
357   legend.position = "none",
358   text = element_text(size = 14)
359 ) +
360 geom_boxplot(outlier.shape = NA, alpha = 0.7) +
361 geom_point(
362   position = position_jitterdodge(
363     jitter.height = 0,
364     jitter.width = 0.5,
365     dodge.width = .1
366   ),
367   shape = 21,
368   alpha = 0.4
369 )#+
370 # coord_cartesian(ylim = c(0, 8))
371 print(q)
372 dev.copy(
373   svg,
374   file = paste0(filename, "_totalcellheight.svg"),
375   width = 4,
376   height = 4
377 )
378 dev.off ()
379
380 #to plot boxplots for Height of cell vs locfit data-----
381 r <-
382 ggplot(data = locfit.model,
383       aes(
384         x = factor(Cell_height),

```



```

385     y = fitvalues,
386     fill = Genotype
387   )) +
388   theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
389         14)) +
390   labs(title = "E-cadherin levels at various Cell Heights", x = "Normalized Height", y = "E-cadherin
391 Levels(in AU)") +
392   geom_boxplot(position = position_dodge(width = 0.8), outlier.shape = NA) +
393   scale_y_continuous(labels = comma) +
394   geom_point(
395     alpha = 0.4,
396     position = position_jitterdodge(jitter.width = .2, dodge.width = 0.8),
397     shape = 21
398   )#+
399 # coord_cartesian(ylim = c(0, 1750))
400 print(r)
401 dev.copy(
402   svg,
403   file = paste0(filename, "_locfit_boxplot.svg"),
404   width = 4,
405   height = 4
406 )
407 dev.off ()
408
409 #to plot lineplots for cell height vs Ecad levels-----
410 s <-
411 ggplot(
412   data = graph.lines,
413   aes(
414     x = Stackheight,
415     y = Mean,
416     group = Uniquecellname,
417     fill = Genotype,
418     color = Genotype
419   )
420 ) +
421 theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
422         14)) +
423 labs(title = "Height vs Ecad Levels",
424       x = "Height(in " ~ (mu * m),
425       y = "E-cadherin Levels(in AU)") +
426 scale_y_continuous(labels = comma) +
427 geom_point(position = position_jitter(width = .1),
428           shape = 21,
429           alpha = 0.4) +
430 geom_line(alpha = 0.4)#+
431 # coord_cartesian(ylim = c(0, 1750))
432 print(s)

```

```

433 dev.copy(
434   svg,
435   file = paste0(filename, "_Rawheight.svg"),
436   width = 4,
437   height = 4
438 )
439 dev.off ()
440
441 #to plot lineplots for Normalized cell height vs Ecad levels-----
442 u <-
443 ggplot(
444   data = graph.lines,
445   aes(
446     x = Normalizedheight,
447     y = Mean,
448     group = Uniquecellname,
449     fill = Genotype,
450     color = Genotype
451   )
452 ) +
453 theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
454   14)) +
455 labs(title = "Normalized Height vs Ecad Levels", x = "Normalized Height", y = "E-cadherin Levels(in
456 AU)") +
457 geom_point(shape = 21, alpha = 0.4) +
458 scale_y_continuous(labels = comma) +
459 geom_line(alpha = 0.4) +
460 #coord_cartesian(ylim = c(0, 1750))#+
461 facet_grid(facets = . ~ Boundary)
462 print(u)
463
464 dev.copy(
465   svg,
466   file = paste0(filename, "_Normalized_height.svg"),
467   width = 4,
468   height = 4
469 )
470 dev.off ()
471
472
473 #to plot lineplots for Normalized cell height vs Normalized Ecad levels-----
474 t <-
475 ggplot(
476   data = graph.lines,
477   aes(
478     x = Normalizedheight,
479     y = NormalizedEcad,
480     group = Uniquecellname,

```

```

481     fill = Genotype
482     # color = Genotype
483   )
484 ) +
485 theme(
486   plot.title = element_text(hjust = 0.5),
487   text = element_text(size = 14),
488   legend.position = "none",
489   axis.text.x = element_text(angle = 45, hjust = 1)
490 ) +
491 scale_y_continuous(labels = comma) +
492 labs(title = "E-Cadherin Distribution", x = "Normalized Height", y = "Normalized E-cadherin Levels")
493 +
494 # stat_smooth(aes(
495 #   x = Normalizedheight,
496 #   y = NormalizedEcad,
497 #   group = Genotype,
498 #   fill = Genotype,
499 #   color = Genotype
500 # ),method='lm')+
501 geom_bin2d(bins = 15, alpha = 0.3) +
502 geom_point(shape = 21, alpha = 0.15) + facet_grid(. ~ Boundary) #+
503
504 # stat_density_2d(aes(colour = Genotype), geom="density_2d", alpha= 0.2)
505 print(t)
506
507 dev.copy(
508   svg,
509   file = paste0(filename, "_Normalized height Normalized Ecad.svg"),
510   width = 4,
511   height = 4
512 )
513 dev.off ()
514
515
516
517 #to plot apical perimeter vs genotype-----
518 b <-
519 ggplot(data = apical.perimeter, aes(x = Genotype, y = Length, fill = Genotype)) +
520 theme(
521   plot.title = element_text(hjust = 0.5),
522   text = element_text(size = 14),
523   legend.position = "none",
524   axis.text.x = element_text(angle = 45, hjust = 1)
525 ) +
526 labs(title = "Apical Perimeter of Cells",
527       x = "Genotype",
528       y = "Length(in" ~ (mu * m)) +

```

```

529 geom_boxplot(outlier.shape = NA, alpha = 0.7) +
530 scale_y_continuous(labels = comma) +
531 theme(
532   plot.title = element_text(hjust = 0.5),
533   legend.position = "none",
534   text = element_text(size = 14)
535 ) +
536 geom_point(position = position_jitter(width = .1),
537            shape = 21,
538            alpha = 0.4)#+
539 # coord_cartesian(ylim = c(40, 120))
540 print(b)
541 dev.copy(
542   svg,
543   file = paste0(filename, "_apical perimeter.svg"),
544   width = 4,
545   height = 4
546 )
547 dev.off ()
548
549
550
551 # multiplot graphs-----
552 # To plot multiple graphs in one page
553
554 grid.newpage()
555 pushViewport(viewport(layout = grid.layout(17, 6)))
556 vplayout <- function(x, y)
557   viewport(layout.pos.row = x, layout.pos.col = y)
558 print(r, vp = vplayout(1:6, 2:6))
559 print(p, vp = vplayout(7:11, 1:2))
560 print(q, vp = vplayout(7:11, 3:4))
561 print(b, vp = vplayout(7:11, 5:6))
562 print(t, vp = vplayout(12:17, 2:5))
563 dev.copy(
564   svg,
565   file = paste0(filename, "multiplot_all.svg"),
566   width = 10,
567   height = 10
568 )
569 dev.off()
570

```

```
571 # Supplementary File 2- R Script for Figure 2, 3, Figure 2-figure supplement 3, 4
572 #
573 #
574 # This is a program to calculate various parameters for E-cadherin quantifications-----
575 #
576 #
577 #
578 # "rawdata" contains data read from the csv file(provided by user). It should contain the following
579 columns:
580 # 1) Genotype
581 # 2) Cell name
582 # 3) Mean
583 # 4) Length
584 # 5) Boundary -- If you are doing cell wise analysis(put same letter in the whole column in case
585 analysis is cell wise and by boundaries)
586 #
587 #
588 # "stack.all" contains stack wise calculated data which includes:
589 # 1) Height at a particular stack
590 # 2) Normalized Height
591 # 3) Normalized Ecad
592 # 4) Absolute Ecad at each stack
593 #
594 # "cell.all" contains cell wise data. Which includes the following:
595 # 1) The total Height of the Cell
596 # 2) Total Ecad in the cell
597 #
598 # "genotype.all" contains genotype wise data which includes the following:
599 # 1) Mean apical perimeter
600 # 2) Median apical perimeter
601 # 3) Mean Total Ecad per cell
602 # 4) Median Total Ecad per cell
603 # 5) Mean Height of the cell
604 # 6) Median Height of the cell
605 # 7) First quantile of Height of cell
606 # 8) Third quantile of height of cell
607 #
608 # "lofit.model" contains the lofit data and the height at which the levels were predicted.
609 #
610 # "apical.perimeter" contains the apical most stack used for apical perimeter analysis
611 #
612 # "linear.reg.noise" contains number of cells/percentage cells showing abnormal polarized
613 distribution
614 #
615 # "linear.reg.whole" contains linear regression for both siblings and mutants
616 #
617 # The program also classifies the cells according to the cell height.-- Tall cells, medium cells, short
618 cells
```

```

619 #
620 #
621 #
622
623 #clears memory-----
624 rm(list = ls())
625
626 #sets a working directory-----
627 #This is where all your graphs will be saved
628
629 setwd(
630   "C:/User/Desktop/"
631 )
632
633 #libraries required for the program to work.-----
634 # List of libraries required for the program to run
635 pkg <- c("scales", "locfit", "tidyr", "purrr", "dplyr", "ggplot2", "broom",
636         "data.table", "RCurl", "stringr", "grid", "gridExtra", "RColorBrewer")
637
638 # Check if the libraries are installed and install libraries not present
639 new.pkg <- pkg[!(pkg %in% installed.packages())]
640 if (length(new.pkg)) {
641   install.packages(new.pkg)
642 }
643 rm("pkg", "new.pkg")
644
645 #load libraries in the memory
646 library(scales)
647 library(locfit)
648 library(tidyr)
649 library(purrr)
650 library(dplyr)
651 library(ggplot2)
652 library(data.table)
653 library(RCurl)
654 library(stringr)
655 library(grid)
656 library(gridExtra)
657 library(RColorBrewer)
658
659 #stops warnings from being displayed
660 options(warn = -1)
661
662
663
664 # read raw data-----
665 # the file should have the following components:
666 # Genotype      Cell name      Area      Mean      Boundary

```

```

667
668 #Folder and file names where your input file is saved
669
670 Folderadd <-
671 "C:/User/Desktop/"
672 filename <-
673 "aPKC mut"
674 format <- ".csv"
675 fulladdress <- paste(Folderadd, filename, format, sep = "")
676
677 rawdata <-
678 read.csv(fulladdress,
679         header = TRUE,
680         sep = ",") %>% na.omit()
681
682 # data arranging-----
683
684 #To have Genotype and Cell name, Boundary order as in the file (Otherwise it comes alphabetically
685 which may not be desirable)
686 rawdata$Genotype <-
687 factor(rawdata$Genotype, levels = unique(rawdata$Genotype))
688 rawdata$Cell.name <-
689 factor(rawdata$Cell.name, levels = unique(rawdata$Cell.name))
690 rawdata$Boundary <-
691 factor(rawdata$Boundary, levels = unique(rawdata$Boundary))
692
693
694 #group by genotype and cell name calculate absolute ecad for each stack, and nest the data to to
695 help calculations on groups
696 by_cellname <- rawdata %>%
697   group_by(Genotype, Cell.name) %>%
698   mutate(AbsoluteEcad = Mean * Area) %>%
699   nest()
700
701
702 #function to find height
703 rawheight <- function(df) {
704   height <- vector("double", nrow(df))
705   for (i in seq_along(df$Mean)) {
706     height[i] <- (i - 1) * 0.28 #the height of the z stack
707   }
708   height
709 }
710
711
712 #function to normalize height
713 normheightfun <- function(df) {
714   nheight <- vector("double", nrow(df))

```

```

715   maxh <- nrow(df)
716   minh <- 1
717   for (i in seq_along(df$Mean)) {
718     nheight[i] <- ((i) - minh) / (maxh - minh)
719   }
720   nheight
721 }
722
723 #function to normalize ecad levels
724 normecadfun <- function(df) {
725   necad <- vector("double", nrow(df))
726   maxe <- max(df$Mean)
727   mine <- min(df$Mean)
728   for (i in seq_along(df$Mean)) {
729     necad[i] <- (df$Mean[i] - mine) * 100 / (maxe - mine)
730   }
731   necad
732 }
733
734 #dataframe which now has height per stack, normalized height, normalized ecad per genotype-cell
735 name
736 stack.all <- by_cellname %>%
737   mutate(
738     Stackheight = data %>% map(rawheight),
739     Normalizedheight = data %>% map(normheightfun),
740     NormalizedEcad = data %>% map(normecadfun),
741     Uniquecellname = paste(Genotype, Cell.name)
742   ) %>% unnest()
743
744
745 #to know the total ecad in a cell and height of a cell
746 cell.all <- stack.all %>%
747   group_by(Genotype, Cell.name) %>%
748   summarize(Total_Ecad = sum(AbsoluteEcad),
749             Height_of_cell = max(Stackheight))
750
751 #to fit linear regression to find the abnormal polarized distribution-----
752
753 #function for linear regression
754
755 linearfit.noise <- function(df) {
756
757   # To give specific colors in linear regressions, a column called colour1 can be added to input file.
758   # all "colourcodes" lines can then be uncommented.
759
760   # colourcodes <- df %>% distinct(colour1)
761   # colourcodes <- as.character(colourcodes[['colour1']])
762   # print(colourcodes)

```



```

763
764
765 # select sibling data for linear regression
766
767 df.lm <- df %>% filter(str_detect(df$Genotype, "sib"))
768
769 # fit linear regression on the sibling data
770 fit <- lm(NormalizedEcad ~ Normalizedheight, data = df.lm)
771
772 # find the upper and lower prediction limits at 99% prediction level
773 pred <-
774   as.data.frame(predict(fit, df, level = 0.99, interval = "prediction"))
775
776 # check if the data is within the prediction limits
777 # TRUE= point is within the predction limit
778 # FALSE = point lies outside the prediction limit
779
780 dat <-
781   data.frame(df,
782     Distribution = df$NormalizedEcad <= pred$upr &
783     df$NormalizedEcad >= pred$lwr)
784
785 # Replace TRUE with normal and FALSE with Abnormal
786
787 dat <- dat %>% mutate(Distribution=replace(Distribution, Distribution=="TRUE", "Normal"))
788 dat <- dat %>% mutate(Distribution=replace(Distribution, Distribution=="FALSE", "Abnormal"))
789
790 # plot graph for the linear regression fit data showing the prediction limits according to the siblings
791
792 plot.lm <-
793   ggplot(data = dat, aes(
794     x = dat$Normalizedheight,
795     y = dat$NormalizedEcad
796   )) +
797   # scale_colour_manual(values = colourcodes) +
798   coord_cartesian(ylim = c(0, 100)) +
799   theme(
800     plot.title = element_text(hjust = 0.5),
801     text = element_text(size = 14),
802     axis.text.x = element_text(angle = 45, hjust = 1)
803   ) +
804   labs(title = "E-Cadherin Distribution", x = "Normalized Height", y = "Normalized E-cadherin Levels")
805 +
806   stat_smooth(method = 'lm',
807     # color = colourcodes[1],
808     se = FALSE
809   ) +
810   geom_ribbon(

```

```

811     data = pred,
812     aes(ymin = lwr, ymax = upr),
813     # fill = colourcodes[1],
814     alpha = 0.2
815   ) +
816   geom_point(aes(colour = Genotype), alpha = 0.4)
817 print(plot.lm)
818
819
820 # save the plot
821 dev.copy(svg,
822   file = paste0(
823     filename,
824     "_",
825     df$Genotype[1],
826     "_linear_regrs_sibling_pred_interval.svg"
827   ))
828 dev.off()
829
830 # Calculate abnormal vs normal distribution numbers and percentage
831 dat.summary <-
832   dat %>% group_by(Genotype, Distribution) %>% distinct(Uniquecellname) %>%
833   arrange(Distribution)
834 dat.summary <-
835   dat.summary %>% ungroup() %>% group_by(Genotype, Uniquecellname) %>% slice(1)
836 dat.summary <-
837   dat.summary %>% group_by(Genotype, Distribution) %>% summarize(Count =
838     n())
839 dat.summary2 <-
840   dat.summary %>% ungroup() %>% group_by(Genotype) %>% summarize(total =
841     sum(Count))
842 dat.summary3 <- full_join(dat.summary, dat.summary2)
843 dat.summary3 <-
844   dat.summary3 %>% mutate(Percentage = ((Count / total) * 100))
845 dat.summary3
846 }
847 # show_col(hue_pal()(4)) #to show color codes
848
849 # Group according the boundary types to get values for boudnaries only (eg. Clone-Clone)
850 linear.reg.input <- stack.all %>%
851   group_by(Boundary) %>%
852   nest()
853
854 # to get table for all normal vs abnormal distribution according to boundary type
855 linear.reg.noise <- linear.reg.input %>%
856   mutate(lmfit = data %>% map(linearfit.noise)) %>% select(-data) %>% unnest(lmfit)
857
858 write.csv(

```

```

859 linear.reg.noise,
860 file = paste0(filename, "_abnormal_by_linear_regression.csv"),
861 row.names = FALSE
862 )
863
864
865 # function to check if linear regressions are close by in both mutants and siblings
866
867 linearfit.whole <- function(df) {
868   # colourcodes <- df %>% distinct(colour1)
869   # colourcodes <- as.character(colourcodes[['colour1']])
870
871   # plot regressions for bith siblings and mutants
872   plot.lm <-
873     ggplot(data = df,
874            aes(
875              x = df$Normalizedheight,
876              y = df$NormalizedEcad,
877              colour = Genotype
878            )) +
879     # scale_colour_manual(values = colourcodes) +
880     coord_cartesian(ylim = c(0, 100)) +
881     theme(
882       plot.title = element_text(hjust = 0.5),
883       text = element_text(size = 14),
884       axis.text.x = element_text(angle = 45, hjust = 1)
885     ) +
886     labs(title = "E-Cadherin Distribution", x = "Normalized Height", y = "Normalized E-cadherin Levels")
887   +
888     stat_smooth(method = 'lm', se = FALSE) +
889     geom_point(aes(), alpha = 0.4)
890   print(plot.lm)
891
892   # save the plot
893   dev.copy(svg, file = paste0(filename, df$Genotype[1], "_linear_regrs_whole.svg"))
894   dev.off()
895
896   # fit linear regression in both sibling and mutant
897
898   fit <- lm(NormalizedEcad ~ Normalizedheight, data = df)
899   radjsq <- summary(fit)$adj.r.squared
900   rsq <- summary(fit)$r.squared
901   slope <- coef(summary(fit))["Normalizedheight", "Estimate"]
902   intercept <- coef(summary(fit))["(Intercept)", "Estimate"]
903   tablefit <- data.frame(radjsq, rsq, intercept, slope)
904   tablefit <-
905     tablefit %>% rename("Adjusted R Squared" = radjsq, "R Squared" = rsq)
906   tablefit

```

```

907 }
908
909 # Make table according with linear regressions fit according to boundary types
910
911 linear.reg.whole <- linear.reg.input %>%
912   mutate(lmfit = data %>% map(linearfit.whole)) %>% select(-data) %>% unnest(lmfit)
913
914 #write csv for linear fit for Normalized ecad and normalized height-----
915
916 write.csv(
917   linear.reg.whole,
918   file = paste0(filename, "_genotypewise_linearfit_coefficients.csv"),
919   row.names = FALSE
920 )
921
922
923 #locfit -----
924 # locfit will fit the data cell wise and predict values at given cell heights
925
926 #nest the data in stack.all with normalized height to use in locfit for prediction
927 locfit.input <- stack.all %>%
928   group_by(Genotype, Cell.name) %>%
929   nest()
930
931 # locfit.input$data[[1]]
932
933
934 #cell heights chosen to interpolate ecad levels
935 predictvalues <- c(0.1, 0.3, 0.5, 0.7, 0.9)
936
937 #To calculate the y-axis limits in locfit graphs
938 ylimmax <- max(rawdata$Mean) + 250
939 ylimmin <- min(rawdata$Mean) - 100
940
941
942 #function to fit locfit
943 locfit_model <- function(df) {
944   fit <- locfit(Mean ~ Normalizedheight, data = df)
945   plot(
946     fit,
947     get.data = TRUE,
948     ylim = c(ylimmin, ylimmax),
949     xlim = c(0, 1),
950     xlab = "Normalized Cell Height",
951     ylab = "Ecad Level(in AU)"
952   )
953   par(new = TRUE) #To superimpose graphs
954   predictions <- predict(fit, predictvalues)

```

```

955
956 }
957
958 #to plot the locfit graphs
959 #to start a new graph page and close all old ones,
960 # so that new graphs don't get superimposed on the old ones
961 plot.new()
962 dev.off()
963 plot.new()
964
965
966
967
968 #add locfit data to grouped cells and add cell height(on which prediction is made) to each cell
969 locfit.model <- locfit.input %>%
970   mutate(fitvalues = data %>% map(locfit_model)) %>%
971   unnest(fitvalues) %>%
972   group_by(Genotype, Cell.name) %>%
973   mutate(Cell_height = predictvalues)
974 locfit.model <- locfit.model %>% group_by(Genotype, Cell.name)
975
976 #to save graph for locfit data
977 dev.copy(
978   svg,
979   file = paste0(filename, "_locfitgraphs.svg"),
980   width = 4,
981   height = 4
982 )
983
984 dev.off ()
985
986
987 #for creating groups for each cell for ggplot graph of lines -----
988
989 graph.lines <- stack.all %>% mutate (Uniquecellname = paste(Genotype, Cell.name))
990
991
992 #to find out the apical perimeter of the cell
993
994 apical.perimeter <- graph.lines %>%
995   group_by(Genotype, Cell.name) %>%
996   slice(1) %>% select(Genotype, Cell.name, Length)
997
998 cell.all <- cell.all %>% full_join(apical.perimeter, by = c("Genotype", "Cell.name"))
999
1000 cell.all <- rename(cell.all, "Apical Perimeter" = Length)
1001
1002

```

```

1003 # to calculate Genotype wise parameters-----
1004 genotype.peri <- apical.perimeter %>% group_by(Genotype) %>%
1005   summarise(Mean_peri = mean(Length),
1006             Median_peri = median(Length))
1007
1008
1009 genotype.summary <- cell.all %>% group_by(Genotype) %>%
1010   summarise(
1011     Mean_TotalEcad = mean(Total_Ecad),
1012     Median_TotalEcad = median(Total_Ecad),
1013     Mean_Heightofcell = mean(Height_of_cell),
1014     Median_Heightofcell = median(Height_of_cell),
1015     First_quantile_Height = quantile(Height_of_cell, prob = 0.25),
1016     Third_quantile_Height = quantile(Height_of_cell, prob = 0.75),
1017     count = n()
1018   )
1019
1020 #final dataframe for all calculated parameters by merging the above two data frames
1021 genotype.all <- genotype.peri %>% full_join(genotype.summary, by = "Genotype")
1022
1023 #filter out tall cells or short cells-----
1024
1025 # This part of the program will not run if the following names are in the name of the file
1026 # (so that the same program can be run on different hieghted cells without really re- filtering the
1027 cells according to the height again)
1028 namecontains <- c("tall_cells", "short_cells", "median_cells")
1029
1030 if (!any(str_detect(filename, namecontains))) {
1031
1032
1033   #filter tall cells
1034   sibthirdquantile <- genotype.summary %>% filter(str_detect(genotype.summary$Genotype, "sib"))
1035   %>% select(Third_quantile_Height)
1036   sibfirstquantile <- genotype.summary %>% filter(str_detect(genotype.summary$Genotype, "sib"))
1037   %>% select(First_quantile_Height)
1038
1039   tall_cells_cellwise <- cell.all %>%
1040     full_join(genotype.all, by = "Genotype") %>% filter(Height_of_cell >
1041 sibthirdquantile$Third_quantile_Height) %>% select(Genotype, Cell.name, Total_Ecad,
1042 Height_of_cell)
1043
1044   tall_cells_all <- stack.all %>%
1045     semi_join(tall_cells_cellwise, by = c("Genotype", "Cell.name"))
1046
1047
1048 #write csv with tall cells filtered out
1049 write.csv(
1050   tall_cells_cellwise,

```

```

1051     file = paste0(filename, "_tall_cells_cellwise.csv"),
1052     row.names = FALSE
1053 )
1054 write.csv(
1055     tall_cells_all,
1056     file = paste0(filename, "_tall_cells_all.csv"),
1057     row.names = FALSE
1058 )
1059
1060 #filter short cells
1061 short_cells_cellwise <- cell.all %>%
1062     full_join(genotype.all, by = "Genotype") %>% filter(Height_of_cell <
1063 sibfirstquantile$First_quantile_Height) %>% select(Genotype, Cell.name, Total_Ecad, Height_of_cell)
1064
1065 short_cells_all <- stack.all %>%
1066     semi_join(short_cells_cellwise, by = c("Genotype", "Cell.name"))
1067 #write csv with short cells filtered out
1068 write.csv(
1069     short_cells_cellwise,
1070     file = paste0(filename, "_short_cells_cellwise.csv"),
1071     row.names = FALSE
1072 )
1073 write.csv(
1074     short_cells_all,
1075     file = paste0(filename, "_short_cells_all.csv"),
1076     row.names = FALSE
1077 )
1078
1079
1080 #filter median heighted cells
1081 median_cells_cellwise <- cell.all %>%
1082     full_join(genotype.all, by = "Genotype") %>% filter(Height_of_cell >=
1083 sibfirstquantile$First_quantile_Height & Height_of_cell <= sibthirdquantile$Third_quantile_Height)
1084 %>% select(Genotype, Cell.name, Total_Ecad, Height_of_cell)
1085
1086 median_cells_all <- stack.all %>%
1087     semi_join(median_cells_cellwise, by = c("Genotype", "Cell.name"))
1088
1089 #write csv with short cells filtered out
1090 write.csv(
1091     median_cells_cellwise,
1092     file = paste0(filename, "_median_cells_cellwise.csv"),
1093     row.names = FALSE
1094 )
1095 write.csv(
1096     median_cells_all,
1097     file = paste0(filename, "_median_cells_all.csv"),
1098     row.names = FALSE

```

```

1099 )
1100
1101 }
1102
1103 #writing csv-----
1104
1105
1106 #write csv with height, normalized height,absolute ecad in each stack
1107 write.csv(
1108   stack.all,
1109   file = paste0(filename, "_stackwise_calculations.csv"),
1110   row.names = FALSE
1111 )
1112
1113 #write csv with cell level parameters
1114 write.csv(
1115   cell.all,
1116   file = paste0(filename, "_cellwise_calculations.csv"),
1117   row.names = FALSE
1118 )
1119
1120 #write csv with locfit fitdata
1121 write.csv(locfit.model,
1122           file = paste0(filename, "_locfitdata.csv"),
1123           row.names = FALSE)
1124
1125 #write csv with genotype level parameters
1126 write.csv(
1127   genotype.all,
1128   file = paste0(filename, "_genotypewise_calculations.csv"),
1129   row.names = FALSE
1130 )
1131
1132 #graphs-----
1133
1134 #to plot boxplots for Total Ecad vs Genotype-----
1135 p <-
1136 ggplot(data = cell.all, aes(x = Genotype, y = Total_Ecad, fill = Genotype)) +
1137 theme(
1138   plot.title = element_text(hjust = 0.5),
1139   text = element_text(size = 14),
1140   legend.position = "none",
1141   axis.text.x = element_text(angle = 45, hjust = 1)
1142 ) +
1143 labs(title = "Total E-cadherin", x = "Genotype", y = "E-cadherin Levels(in AU)") +
1144 scale_y_continuous(labels = comma) +
1145 theme(
1146   plot.title = element_text(hjust = 0.5),

```



```

1147     legend.position = "none",
1148     text = element_text(size = 14)
1149 ) +
1150 geom_boxplot(outlier.shape = NA, alpha = 0.7) +
1151 geom_point(position = position_jitter(width = .1),
1152            shape = 21,
1153            alpha = 0.4)#+
1154 # coord_cartesian(ylim = c(0, 800000))
1155 print(p)
1156 dev.copy(
1157     svg,
1158     file = paste0(filename, "_Total Ecad.svg"),
1159     width = 4,
1160     height = 4
1161 )
1162 dev.off ()
1163
1164
1165
1166 #to plot boxplots for Height of cell vs Genotype-----
1167 q <-
1168 q <-
1169 ggplot(data = cell.all, aes(x = Genotype, y = Height_of_cell, fill = Genotype)) +
1170 theme(
1171     plot.title = element_text(hjust = 0.5),
1172     text = element_text(size = 14),
1173     legend.position = "none",
1174     axis.text.x = element_text(angle = 45, hjust = 1)
1175 ) +
1176 labs(title = "Height of Cells",
1177      x = "Genotype",
1178      y = expression(paste("Height(in ", mu, "m)"))) +
1179 scale_y_continuous(labels = comma) +
1180 theme(
1181     plot.title = element_text(hjust = 0.5),
1182     legend.position = "none",
1183     text = element_text(size = 14)
1184 ) +
1185 geom_boxplot(outlier.shape = NA, alpha = 0.7) +
1186 geom_point(
1187     position = position_jitterdodge(
1188         jitter.height = 0,
1189         jitter.width = 0.5,
1190         dodge.width = .1
1191     ),
1192     shape = 21,
1193     alpha = 0.4
1194 )#+

```

```

1195 # coord_cartesian(ylim = c(0, 8))
1196 print(q)
1197 dev.copy(
1198   svg,
1199   file = paste0(filename, "_totalcellheight.svg"),
1200   width = 4,
1201   height = 4
1202 )
1203 dev.off ()
1204
1205 #to plot boxplots for Height of cell vs locfit data-----
1206 r <-
1207   ggplot(data = locfit.model,
1208     aes(
1209       x = factor(Cell_height),
1210       y = fitvalues,
1211       fill = Genotype
1212     )) +
1213     theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
1214       14)) +
1215     labs(title = "E-cadherin levels at various Cell Heights", x = "Normalized Height", y = "E-cadherin
1216 Levels(in AU)") +
1217     geom_boxplot(position = position_dodge(width = 0.8), outlier.shape = NA) +
1218     scale_y_continuous(labels = comma) +
1219     geom_point(
1220       alpha = 0.4,
1221       position = position_jitterdodge(jitter.width = .2, dodge.width = 0.8),
1222       shape = 21
1223     )#+
1224 # coord_cartesian(ylim = c(0, 1750))
1225 print(r)
1226 dev.copy(
1227   svg,
1228   file = paste0(filename, "_locfit_boxplot.svg"),
1229   width = 4,
1230   height = 4
1231 )
1232 dev.off ()
1233
1234 #to plot lineplots for cell height vs Ecad levels-----
1235 s <-
1236   ggplot(
1237     data = graph.lines,
1238     aes(
1239       x = Stackheight,
1240       y = Mean,
1241       group = Uniquecellname,
1242       fill = Genotype,

```

```

1243     color = Genotype
1244   )
1245 ) +
1246 theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
1247                               14)) +
1248 labs(title = "Height vs Ecad Levels",
1249       x = "Height(in " ~ (mu * m),
1250       y = "E-cadherin Levels(in AU)") +
1251 scale_y_continuous(labels = comma) +
1252 geom_point(position = position_jitter(width = .1),
1253            shape = 21,
1254            alpha = 0.4) +
1255 geom_line(alpha = 0.4)#+
1256 # coord_cartesian(ylim = c(0, 1750))
1257 print(s)
1258 dev.copy(
1259   svg,
1260   file = paste0(filename, "_Rawheight.svg"),
1261   width = 4,
1262   height = 4
1263 )
1264 dev.off ()
1265
1266 #to plot lineplots for Normalized cell height vs Ecad levels-----
1267 u <-
1268 ggplot(
1269   data = graph.lines,
1270   aes(
1271     x = Normalizedheight,
1272     y = Mean,
1273     group = Uniquecellname,
1274     fill = Genotype,
1275     color = Genotype
1276   )
1277 ) +
1278 theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
1279                               14)) +
1280 labs(title = "Normalized Height vs Ecad Levels", x = "Normalized Height", y = "E-cadherin Levels(in
1281 AU)") +
1282 geom_point(shape = 21, alpha = 0.4) +
1283 scale_y_continuous(labels = comma) +
1284 geom_line(alpha = 0.4) +
1285 #coord_cartesian(ylim = c(0, 1750))#+
1286 facet_grid(facets = . ~ Boundary)
1287 print(u)
1288
1289 dev.copy(
1290   svg,

```

```

1291   file = paste0(filename, "_Normalized_height.svg"),
1292   width = 4,
1293   height = 4
1294 )
1295 dev.off ()
1296
1297
1298 #to plot lineplots for Normalized cell height vs Normalized Ecad levels-----
1299 t <-
1300   ggplot(
1301     data = graph.lines,
1302     aes(
1303       x = Normalizedheight,
1304       y = NormalizedEcad,
1305       group = Uniquecellname,
1306       fill = Genotype
1307       # color = Genotype
1308     )
1309   ) +
1310   theme(
1311     plot.title = element_text(hjust = 0.5),
1312     text = element_text(size = 14),
1313     legend.position = "none",
1314     axis.text.x = element_text(angle = 45, hjust = 1)
1315   ) +
1316   scale_y_continuous(labels = comma) +
1317   labs(title = "E-Cadherin Distribution", x = "Normalized Height", y = "Normalized E-cadherin Levels")
1318 +
1319   # stat_smooth(aes(
1320   #   x = Normalizedheight,
1321   #   y = NormalizedEcad,
1322   #   group = Genotype,
1323   #   fill = Genotype,
1324   #   color = Genotype
1325   # ),method='lm')+
1326   geom_bin2d(bins = 15, alpha = 0.3) +
1327   geom_point(shape = 21, alpha = 0.15) + facet_grid(. ~ Boundary) #+
1328
1329 # stat_density_2d(aes(colour = Genotype), geom="density_2d", alpha= 0.2)
1330 print(t)
1331
1332 dev.copy(
1333   svg,
1334   file = paste0(filename, "_Normalized height Normalized Ecad.svg"),
1335   width = 4,
1336   height = 4
1337 )
1338 dev.off ()

```

```

1339
1340
1341 #to plot Percent Abnormal Distribution-----
1342 z <-
1343   ggplot(linear.reg.noise,
1344     aes(x = Genotype, y = Percentage, fill = Distribution)) +
1345   geom_bar(stat = "identity", position = "stack") + scale_fill_brewer(palette = "Dark2") +
1346   scale_y_continuous(labels = comma) +
1347   theme(
1348     plot.title = element_text(hjust = 0.5),
1349     text = element_text(size = 14),
1350     axis.text.x = element_text(angle = 45, hjust = 1)
1351   ) +
1352   labs(title = "Percent Boundaries showing \nAbnormal Distribution", x = "Genotype", y =
1353     "Percentage")
1354
1355
1356 print(z)
1357 dev.copy(
1358   svg,
1359   file = paste0(filename, "_abnormal distribution.svg"),
1360   width = 4,
1361   height = 4
1362 )
1363 dev.off ()
1364
1365 #to plot apical perimeter vs genotype-----
1366 b <-
1367   ggplot(data = apical.perimeter, aes(x = Genotype, y = Length, fill = Genotype)) +
1368   theme(
1369     plot.title = element_text(hjust = 0.5),
1370     text = element_text(size = 14),
1371     legend.position = "none",
1372     axis.text.x = element_text(angle = 45, hjust = 1)
1373   ) +
1374   labs(title = "Apical Perimeter of Cells",
1375     x = "Genotype",
1376     y = "Length(in" ~ (mu * m)) +
1377   geom_boxplot(outlier.shape = NA, alpha = 0.7) +
1378   scale_y_continuous(labels = comma) +
1379   theme(
1380     plot.title = element_text(hjust = 0.5),
1381     legend.position = "none",
1382     text = element_text(size = 14)
1383   ) +
1384   geom_point(position = position_jitter(width = .1),
1385     shape = 21,
1386     alpha = 0.4)#+

```

```
1387 # coord_cartesian(ylim = c(40, 120))
1388 print(b)
1389 dev.copy(
1390   svg,
1391   file = paste0(filename, "_apical perimeter.svg"),
1392   width = 4,
1393   height = 4
1394 )
1395 dev.off ()
1396
1397
1398
1399 # multiplot graphs-----
1400 # To plot multiple graphs in one page
1401
1402 grid.newpage()
1403 pushViewport(viewport(layout = grid.layout(17, 6)))
1404 vplotout <- function(x, y)
1405   viewport(layout.pos.row = x, layout.pos.col = y)
1406 print(r, vp = vplotout(1:6, 2:6))
1407 print(p, vp = vplotout(7:11, 1:2))
1408 print(q, vp = vplotout(7:11, 3:4))
1409 print(b, vp = vplotout(7:11, 5:6))
1410 print(t, vp = vplotout(12:17, 1:4))
1411 print(z, vp = vplotout(12:17, 5:6))
1412 dev.copy(
1413   svg,
1414   file = paste0(filename, "multiplot_all.svg"),
1415   width = 10,
1416   height = 10
1417 )
1418 dev.off()
1419
```

```

1420 # Supplementary File 2- R Script for Figure 4, 5, Figure 4-figure supplement 2
1421 #
1422 #
1423 # This is a program to calculate various parameters for E-cadherin quantifications-----
1424 #
1425 #
1426 #
1427 # "rawdata" contains data read from the csv file(provided by user). It should contain the following
1428 columns:
1429 # 1) Genotype
1430 # 2) Cell name
1431 # 3) Mean
1432 # 4) Length
1433 # 5) Boundary -- If you are doing cell wise analysis(put same letter in the whole column in case
1434 analysis is cell wise and by boundaries)
1435 #
1436 #
1437 # "stack.all" contains stack wise calculated data which includes:
1438 # 1) Height at a particular stack
1439 # 2) Normalized Height
1440 # 3) Normalized Ecad
1441 # 4) Absolute Ecad at each stack
1442
1443 #clears memory-----
1444 rm(list = ls())
1445
1446 #sets a working directory-----
1447 #This is where all your graphs will be saved
1448
1449 setwd(
1450 "D:/Prateek/TIFR/MS Lab/Ecad Gradient/locfit analysis/R files version control/Ecad_project/"
1451 )
1452
1453 #libraries required for the program to work.-----
1454 # List of libraries required for the program to run
1455 pkg <- c("scales", "locfit", "tidyr", "purrr", "dplyr", "ggplot2", "broom",
1456 "data.table", "RCurl", "stringr", "grid", "gridExtra", "RColorBrewer")
1457
1458 # Check if the libraries are installed and install libraries not present
1459 new.pkg <- pkg[!(pkg %in% installed.packages())]
1460 if (length(new.pkg)) {
1461   install.packages(new.pkg)
1462 }
1463 rm("pkg", "new.pkg")
1464
1465 #load libraries in the memory
1466 library(scales)
1467 library(locfit)

```

```

1468 library(tidyr)
1469 library(purrr)
1470 library(dplyr)
1471 library(ggplot2)
1472 library(data.table)
1473 library(RCurl)
1474 library(stringr)
1475 library(grid)
1476 library(gridExtra)
1477 library(RColorBrewer)
1478
1479 #stops warnings from being displayed
1480 options(warn = -1)
1481
1482
1483
1484 # read raw data-----
1485 # the file should have the following components:
1486 # Genotype      Cell name      Area      Mean      Boundary
1487
1488 #Folder and file names where your input file is saved
1489
1490 Folderadd <-
1491 "D:/Prateek/TIFR/MS Lab/Ecad Gradient/locfit analysis/R files version control/Ecad_project/"
1492 filename <-
1493 "Autonomous peri Lgl 05052019 14122018_outliers removed"
1494 format <- ".csv"
1495 fulladdress <- paste(Folderadd, filename, format, sep = "")
1496
1497 rawdata <-
1498   read.csv(fulladdress,
1499           header = TRUE,
1500           sep = ",") %>% na.omit()
1501
1502 # data arranging-----
1503
1504 #To have Genotype and Cell name, Boundary order as in the file (Otherwise it comes alphabetically
1505 which may not be desirable)
1506 rawdata$Genotype <-
1507   factor(rawdata$Genotype, levels = unique(rawdata$Genotype))
1508 rawdata$Cell.name <-
1509   factor(rawdata$Cell.name, levels = unique(rawdata$Cell.name))
1510 rawdata$Boundary <-
1511   factor(rawdata$Boundary, levels = unique(rawdata$Boundary))
1512
1513
1514 #group by genotype and cell name calculate absolute ecad for each stack, and nest the data to to
1515 help calculations on groups

```



```

1516 by_cellname <- rawdata %>%
1517   group_by(Genotype, Cell.name) %>%
1518   mutate(AbsoluteEcad = Mean * Area) %>%
1519   nest()
1520
1521
1522 #function to find height
1523 rawheight <- function(df) {
1524   height <- vector("double", nrow(df))
1525   for (i in seq_along(df$Mean)) {
1526     height[i] <- (i - 1) * 0.28 #the height of the z slice
1527   }
1528   height
1529 }
1530
1531
1532 #function to normalize height
1533 normheightfun <- function(df) {
1534   nheight <- vector("double", nrow(df))
1535   maxh <- nrow(df)
1536   minh <- 1
1537   for (i in seq_along(df$Mean)) {
1538     nheight[i] <- ((i) - minh) / (maxh - minh)
1539   }
1540   nheight
1541 }
1542
1543 #function to normalize ecad levels
1544 normecadfun <- function(df) {
1545   necad <- vector("double", nrow(df))
1546   maxe <- max(df$Mean)
1547   mine <- min(df$Mean)
1548   for (i in seq_along(df$Mean)) {
1549     necad[i] <- (df$Mean[i] - mine) * 100 / (maxe - mine)
1550   }
1551   necad
1552 }
1553
1554 #dataframe which now has height per stack, normalized height, normalized ecad per genotype-cell
1555 name
1556 stack.all <- by_cellname %>%
1557   mutate(
1558     Stackheight = data %>% map(rawheight),
1559     Normalizedheight = data %>% map(normheightfun),
1560     NormalizedEcad = data %>% map(normecadfun),
1561     Uniquecellname = paste(Genotype, Cell.name)
1562   ) %>% unnest()
1563

```

```

1564
1565 #locfit -----
1566 # locfit will fit the data cell wise and predict values at given cell heights
1567
1568 #nest the data in stack.all with normalized height to use in locfit for prediction
1569 locfit.input <- stack.all %>%
1570   group_by(Genotype, Cell.name) %>%
1571   nest()
1572
1573 # locfit.input$data[[1]]
1574
1575
1576 #cell heights chosen to interpolate ecad levels
1577 predictvalues <- c(0.1, 0.3, 0.5, 0.7, 0.9)
1578
1579 #To calculate the y-axis limits in locfit graphs
1580 ylimmax <- max(rawdata$Mean) + 250
1581 ylimmin <- min(rawdata$Mean) - 100
1582
1583
1584 #function to fit locfit
1585 locfit_model <- function(df) {
1586   fit <- locfit(Mean ~ Normalizedheight, data = df)
1587   plot(
1588     fit,
1589     get.data = TRUE,
1590     ylim = c(ylimmin, ylimmax),
1591     xlim = c(0, 1),
1592     xlab = "Normalized Cell Height",
1593     ylab = "Ecad Level(in AU)"
1594   )
1595   par(new = TRUE) #To superimpose graphs
1596   predictions <- predict(fit, predictvalues)
1597
1598 }
1599
1600 #to plot the locfit graphs
1601 #to start a new graph page and close all old ones,
1602 # so that new graphs don't get superimposed on the old ones
1603 plot.new()
1604 dev.off()
1605 plot.new()
1606
1607
1608
1609
1610 #add locfit data to grouped cells and add cell height(on which prediction is made) to each cell
1611 locfit.model <- locfit.input %>%

```

```

1612 mutate(fitvalues = data %>% map(locfit_model)) %>%
1613 unnest(fitvalues) %>%
1614 group_by(Genotype, Cell.name) %>%
1615 mutate(Cell_height = predictvalues)
1616 locfit.model <- locfit.model %>% group_by(Genotype, Cell.name)
1617
1618 #to save graph for locfit data
1619 dev.copy(
1620   svg,
1621   file = paste0(filename, "_locfitgraphs.svg"),
1622   width = 4,
1623   height = 4
1624 )
1625
1626 dev.off ()
1627
1628
1629 #for creating groups for each cell for ggplot graph of lines -----
1630
1631 graph.lines <- stack.all %>% mutate (Uniquecellname = paste(Genotype, Cell.name))
1632
1633
1634
1635
1636 #writing csv-----
1637
1638
1639 #write csv with height, normalized height,absolute ecad in each stack
1640 write.csv(
1641   stack.all,
1642   file = paste0(filename, "_stackwise_calculations.csv"),
1643   row.names = FALSE
1644 )
1645
1646
1647 #write csv with locfit fitdata
1648 write.csv(locfit.model,
1649   file = paste0(filename, "_locfitdata.csv"),
1650   row.names = FALSE)
1651
1652 #graphs-----
1653
1654 #to plot boxplots for Height of cell vs locfit data-----
1655 r <-
1656 ggplot(data = locfit.model,
1657   aes(
1658     x = factor(Cell_height),
1659     y = fitvalues,

```

```

1660         fill = Genotype
1661     )) +
1662     theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
1663         14)) +
1664     labs(title = "E-cadherin levels at various Cell Heights", x = "Normalized Height", y = "E-cadherin
1665 Levels(in AU)") +
1666     geom_boxplot(position = position_dodge(width = 0.8), outlier.shape = NA) +
1667     scale_y_continuous(labels = comma) +
1668     geom_point(
1669     alpha = 0.4,
1670     position = position_jitterdodge(jitter.width = .2, dodge.width = 0.8),
1671     shape = 21
1672     )#+
1673 # coord_cartesian(ylim = c(0, 1750))
1674 print(r)
1675 dev.copy(
1676     svg,
1677     file = paste0(filename, "_locfit_boxplot.svg"),
1678     width = 4,
1679     height = 4
1680 )
1681 dev.off ()
1682
1683 #to plot lineplots for cell height vs Ecad levels-----
1684 s <-
1685 ggplot(
1686     data = graph.lines,
1687     aes(
1688     x = Stackheight,
1689     y = Mean,
1690     group = Uniquecellname,
1691     fill = Genotype,
1692     color = Genotype
1693     )
1694 ) +
1695 theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
1696     14)) +
1697 labs(title = "Height vs Ecad Levels",
1698     x = "Height(in " ~ (mu * m),
1699     y = "E-cadherin Levels(in AU)") +
1700 scale_y_continuous(labels = comma) +
1701 geom_point(position = position_jitter(width = .1),
1702     shape = 21,
1703     alpha = 0.4) +
1704 geom_line(alpha = 0.4)#+
1705 # coord_cartesian(ylim = c(0, 1750))
1706 print(s)
1707 dev.copy(

```

```

1708  svg,
1709  file = paste0(filename, "_Rawheight.svg"),
1710  width = 4,
1711  height = 4
1712  )
1713  dev.off ()
1714
1715  #to plot lineplots for Normalized cell height vs Ecad levels-----
1716  u <-
1717  ggplot(
1718    data = graph.lines,
1719    aes(
1720      x = Normalizedheight,
1721      y = Mean,
1722      group = Uniquecellname,
1723      fill = Genotype,
1724      color = Genotype
1725    )
1726  ) +
1727  theme(plot.title = element_text(hjust = 0.5), text = element_text(size =
1728                                     14)) +
1729  labs(title = "Normalized Height vs Ecad Levels", x = "Normalized Height", y = "E-cadherin Levels(in
1730 AU)") +
1731  geom_point(shape = 21, alpha = 0.4) +
1732  scale_y_continuous(labels = comma) +
1733  geom_line(alpha = 0.4) +
1734  #coord_cartesian(ylim = c(0, 1750))#+
1735  facet_grid(facets = . ~ Boundary)
1736  print(u)
1737
1738  dev.copy(
1739    svg,
1740    file = paste0(filename, "_Normalized_height.svg"),
1741    width = 4,
1742    height = 4
1743  )
1744  dev.off ()
1745
1746
1747  # multiplot graphs-----
1748  # To plot multiple graphs in one page
1749
1750  grid.newpage()
1751  pushViewport(viewport(layout = grid.layout(17, 6)))
1752  vplot <- function(x, y)
1753    viewport(layout.pos.row = x, layout.pos.col = y)
1754  print(r, vp = vplot(1:6, 2:6))
1755  # print(p, vp = vplot(7:11, 1:2))

```

```
1756 # print(q, vp = vplayout(7:11, 3:4))
1757 # print(b, vp = vplayout(7:11, 5:6))
1758 # print(t, vp = vplayout(12:17, 1:4))
1759 # print(z, vp = vplayout(12:17, 5:6))
1760 dev.copy(
1761   svg,
1762   file = paste0(filename, "multiplot_all.svg"),
1763   width = 10,
1764   height = 10
1765 )
1766 dev.off()
1767
```