

Drug-Induced Phenotypes

Jan Sauer

2018-07-12

Contents

1	Introduction	2
2	Define Active Drugs	2
3	KiStem library	8
3.1	Individual Cell Lines	8
3.2	Enrichment Heatmap	37
3.3	Feature Differences	42
3.4	Drug Effect Differences across all Lines	152
4	Clinical Cancer Panel	183
4.1	Drugs Active at All Concentrations	183

1 Introduction

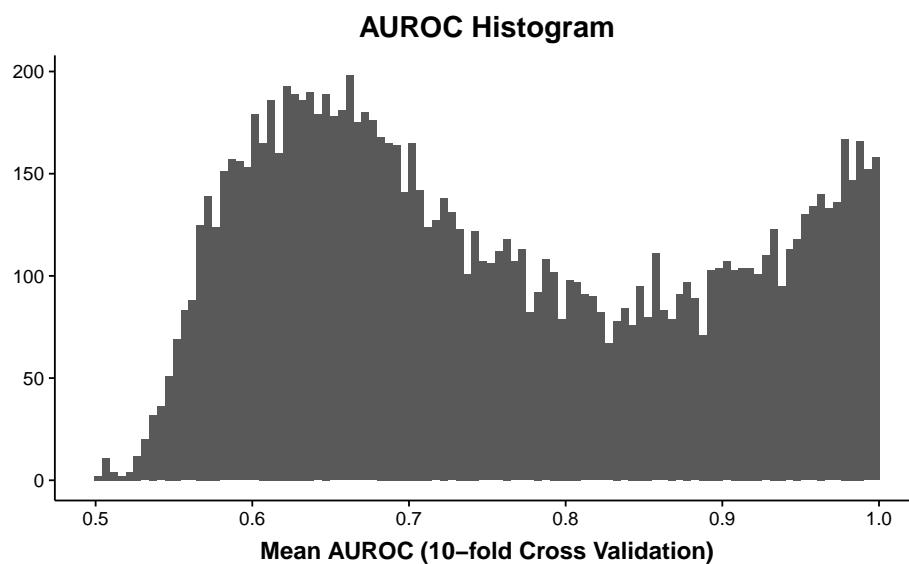
Drug effects are defined as the vector pointing from negative control (DMSO) organoids to treated organoids in feature space. Features are transformed with a PCA and the top 25 components are retained. An SVM with a linear kernel is trained to differentiate DMSO and treated organoids (for each drug individually) and the normal vector to the separating hyperplane is defined as the drug effect vector. It is scaled to be as long as the distance between the DMSO median and the treated organoid median in feature space.

The paper's "story flow" began with the drug-induced phenotypes and then continued onto the organoid viability. However, there are interesting parallels between the two analyses so that I include the lethality of drugs in the annotation of heatmaps here.

2 Define Active Drugs

A histogram of the accuracies (or AUCs to be exact) shows that there are two groups of drugs. The first, normally distributed, group with low AUCs represents the inactive drugs that cannot be sufficiently differentiated from DMSO while the other group represents the active drugs that have significantly different characteristic features than the DMSO organoids and can be separated by a linear SVM hyperplane in feature space.

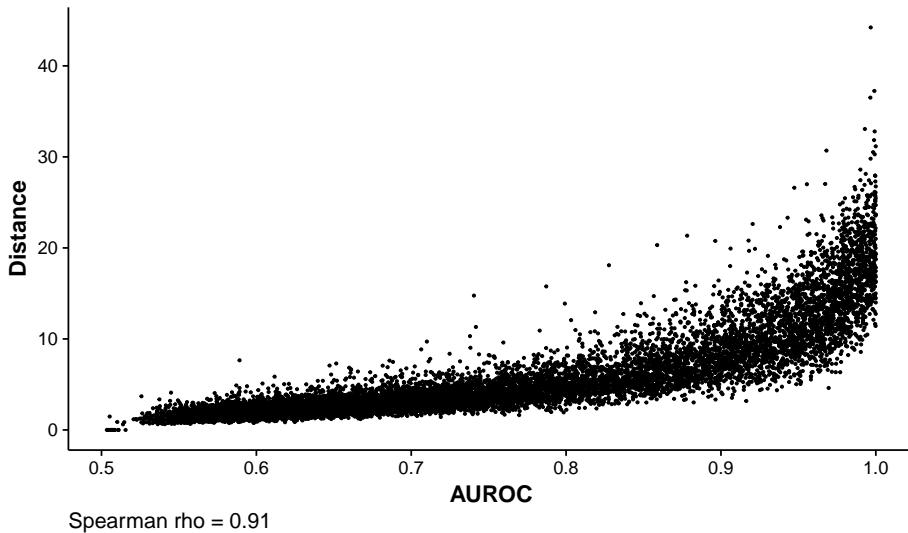
```
ggplot(data = drug_effect_metadata, mapping = aes(x = AUC_Mean)) +  
  geom_histogram(bins = 100) +  
  theme_vignette() + ggtitle("AUROC Histogram") +  
  xlab("Mean AUROC (10-fold Cross Validation)") + ylab("") +  
  if(save_images) ggsave(  
    filename = file.path(img_out_dir, "Histogram_ClassifierAUCs.pdf"),  
    width = 3, height = 3, useDingbats = FALSE)
```



The distance in feature space between drugs and DMSO clouds correlates, as expected, with the AUROC.

Drug-Induced Phenotypes

```
ggplot(data = drug_effect_metadata,
       mapping = aes(x = AUC_Mean, y = Distance)) +
  geom_point(size = 0.5) +
  theme_vignette() + labs(
    caption = paste("Spearman rho =", round(
      cor(drug_effect_metadata$AUC_Mean,
           drug_effect_metadata$Distance, method = "spearman"), 2))) +
  xlab("AUROC") + labs(color = "")
```

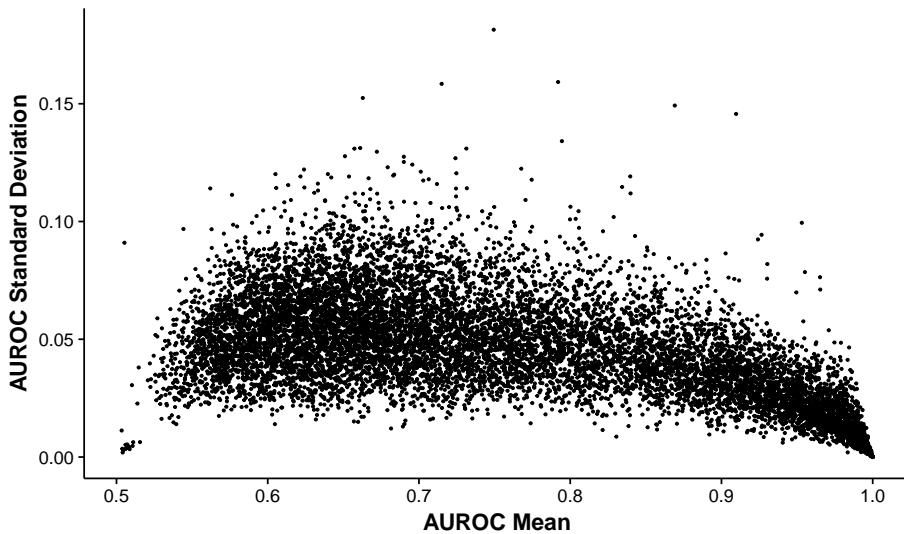


```
if(save_images) ggsave(
  filename = file.path(img_out_dir, "Scatterplot_AUCvsDistance.pdf"),
  width = 3, height = 3, useDingbats = FALSE)
```

The SVM was trained with 10-fold cross validation and the resulting means and standard deviations of the accuracies are shown below.

```
ggplot(data = drug_effect_metadata,
       mapping = aes(x = AUC_Mean, y = AUC_Std)) +
  geom_point(size = 0.5) + theme_vignette() +
  labs(x = "AUROC Mean", y = "AUROC Standard Deviation", color = "")
```

Drug-Induced Phenotypes



```
if(save_images) ggsave(  
  filename = file.path(img_out_dir, "Scatterplot_AUCMeanvsAUCStd.pdf"),  
  width = 3, height = 3, useDingbats = FALSE)
```

Based on the bimodal histogram of active drugs, I define any drug that can be separated with $AUC = 0.85$ from the DMSO controls as “active”.

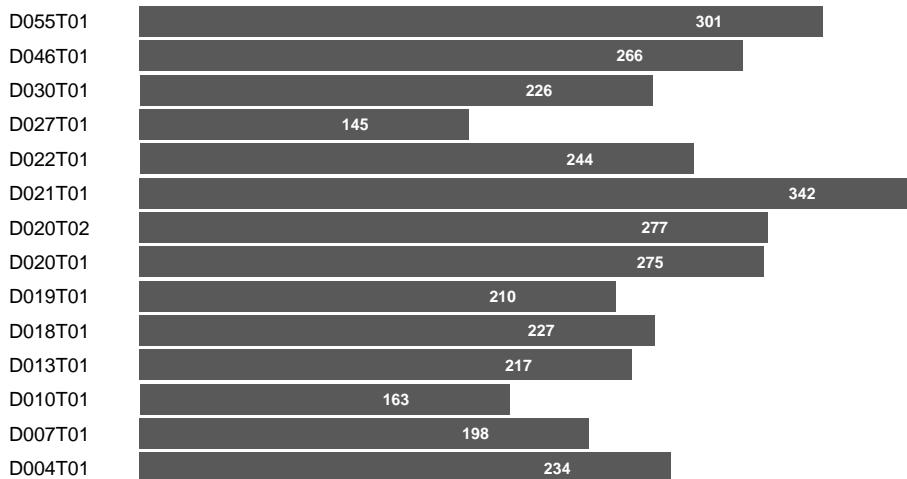
```
auc_thresh = 0.85  
profiles_active = drug_effect_profiles[drug_effect_metadata$AUC_Mean >= auc_thresh, ]  
metadata_active = drug_effect_metadata[drug_effect_metadata$AUC_Mean >= auc_thresh, ]
```

The number of active drugs varies per line.

```
ggplot_df = as.data.frame(table(metadata_active$Line))  
# D054T01 was only imaged with the small library  
ggplot_df = ggplot_df[ggplot_df$Var1 != "D054T01", ]  
ggplot_df$TextLoc = ggplot_df$Freq - 50  
ggplot(data = ggplot_df, mapping = aes(x = Var1, y = Freq)) +  
  geom_col() +  
  geom_text(mapping = aes(x = Var1, y = TextLoc, label = Freq),  
            color = "white", fontface = "bold", size = 3) +  
  theme_vignette() +  
  theme(axis.line = element_blank(), axis.text.x = element_blank(),  
        axis.ticks = element_blank()) +  
  xlab("") + ylab("") + ggtitle("Number of Active Drugs") +  
  coord_flip()
```

Drug-Induced Phenotypes

Number of Active Drugs

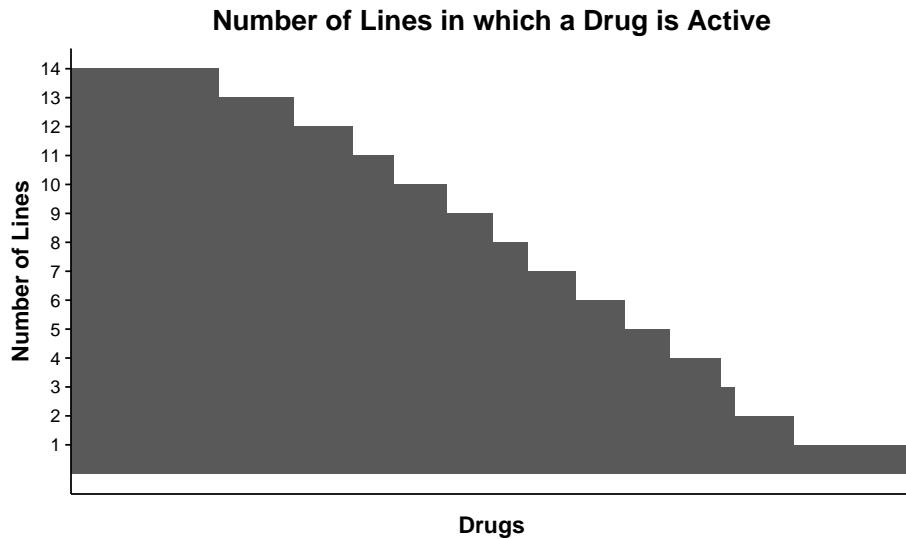


```
if(save_images) ggsave(  
  filename = file.path(img_out_dir, "Barplot_NumActiveDrugs.pdf"),  
  width = 2.5, height = 3)
```

Some drugs are active in all cell lines and others only in very few.

```
activity_table = table(metadata_active$Drug, metadata_active$Line)  
activity_table = activity_table[, colnames(activity_table) != "D054T01"]  
ggplot_df = as.data.frame(sort(rowSums(activity_table > 0), decreasing = TRUE))  
  
colnames(ggplot_df) = "Freq"  
ggplot_df$Var1 = factor(rownames(ggplot_df), levels = rownames(ggplot_df))  
ggplot(data = ggplot_df, mapping = aes(x = Var1, y = Freq)) +  
  geom_col(width = 1) + theme_vignette() + theme(  
    axis.text.x = element_blank(), axis.ticks.x = element_blank()) +  
  xlab("Drugs") + ylab("Number of Lines") +  
  ggtitle("Number of Lines in which a Drug is Active") +  
  scale_y_continuous(breaks = seq_len(length(lines)))
```

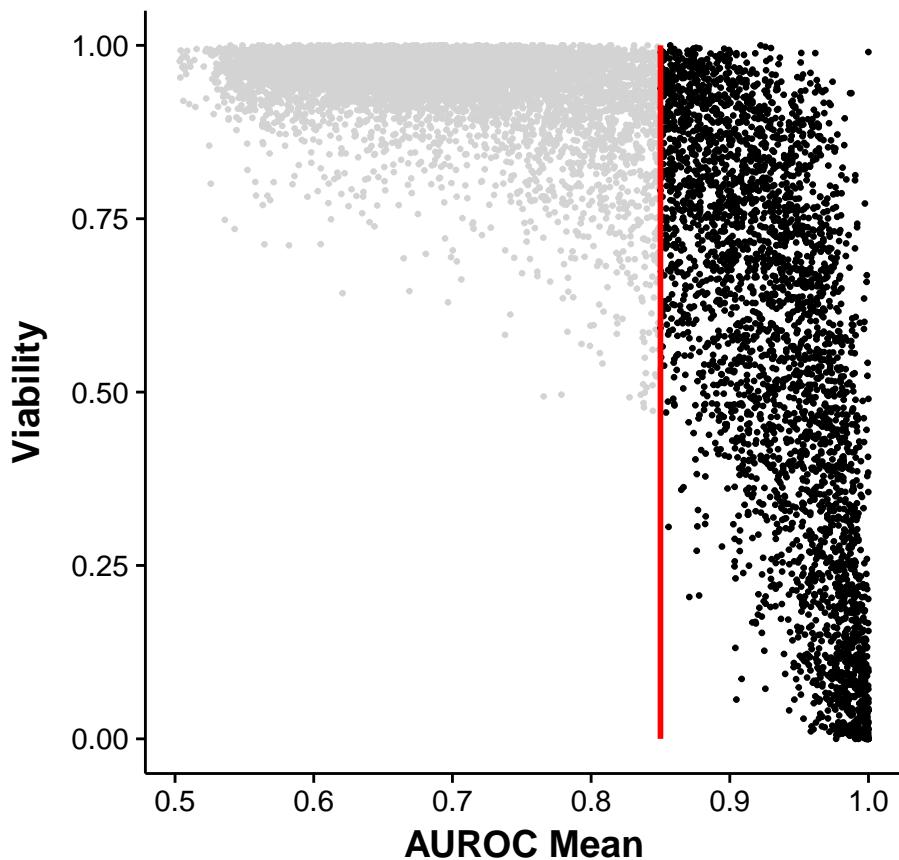
Drug-Induced Phenotypes



A comparison of the viability versus the drug effect shows that all drugs previously identified as lethal are also identified as active. However, a good number of active drugs were previously identified as nonlethal.

```
ggplot(mapping = aes(y = Viability, x = AUC_Mean)) +  
  geom_point(data = drug_effect_metadata, color = "lightgrey", size = 0.5) +  
  geom_point(data = drug_effect_metadata[  
    drug_effect_metadata$AUC_Mean >= auc_thresh, ],  
    size = 0.5) +  
  theme_vignette() + ylab("Viability") + xlab("AUROC Mean") +  
  geom_segment(mapping = aes(  
    y = 0, yend = 1, x = auc_thresh, xend = auc_thresh),  
    color = "red", size = 1) + coord_equal(ratio = 0.5)
```

Drug-Induced Phenotypes

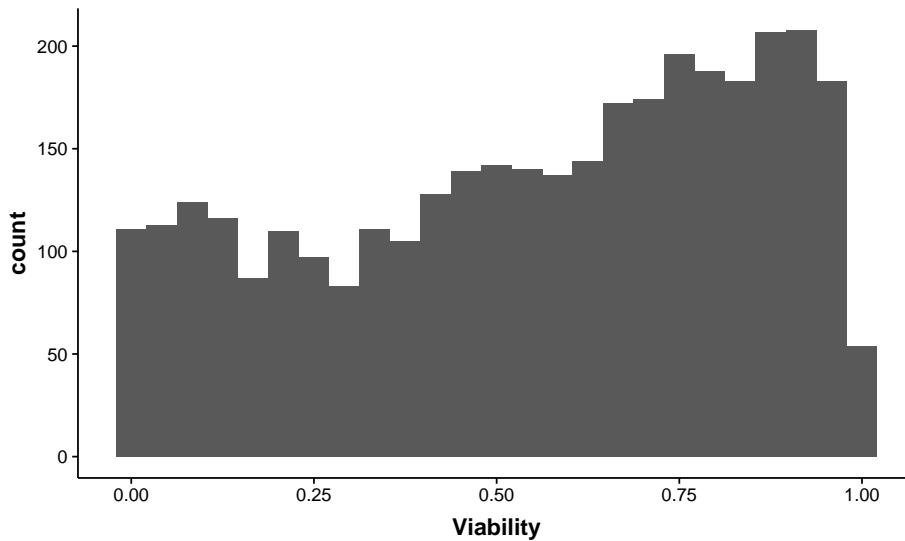


```
if(save_images) ggsave(  
  filename = file.path(img_out_dir, "Scatterplot_ViabilityVsSVMAUROC.pdf"),  
  width = 3, height = 3, useDingbats = FALSE)
```

Plotting a histogram of the viabilities of all “active” drugs, i.e. with $AUROC \geq 0.85$

```
ggplot(data = drug_effect_metadata[  
  drug_effect_metadata$AUC_Mean >= auc_thresh, ]) +  
  geom_histogram(mapping = aes(x = Viability), bins = 25) +  
  theme_vignette()
```

Drug-Induced Phenotypes



3 KiStem library

I analyze the two libraries separately and begin with the KiStem library.

```
sel_indices = (is.na(drug_effect_metadata$Concentration) | (
  drug_effect_metadata$Drug %in% c("Bortezomib", "Irinotecan / SN-38") &
  drug_effect_metadata$Concentration %in% c(0.2, 1))) &
# D054 was only treated with the CCP library
  drug_effect_metadata$Line != "D054T01"
profiles = drug_effect_profiles[sel_indices, ]
metadata = drug_effect_metadata[sel_indices, ]

sel_indices = metadata$AUC_Mean >= auc_thresh
profiles_active = profiles[sel_indices, ]
metadata_active = metadata[sel_indices, ]
```

3.1 Individual Cell Lines

I look at the angles between vectors. The SVM generates a profile for all drugs, including inactive ones. However, because they cannot be properly separated from DMSO, the profiles of these drugs will be largely random, ergo I remove them as noise.

```
killing_anno = rep(NA, nrow(metadata_active))
killing_anno[metadata_active$Viability <= 0.2] = "0.2"
killing_anno[
  metadata_active$Viability > 0.2 &
  metadata_active$Viability <= 0.4] = "0.4"
killing_anno[
  metadata_active$Viability > 0.4 &
  metadata_active$Viability <= 0.6] = "0.6"
```

Drug-Induced Phenotypes

```
killing_anno[
  metadata_active$Viability > 0.6 &
  metadata_active$Viability <= 0.8] = "0.8"
killing_anno[
  metadata_active$Viability > 0.8 &
  metadata_active$Viability <= 1] = "1"

annotation = data.frame(
  "Viability" = killing_anno,
  "Pathway" = SCOPEAnalysis::get_mode_of_action(metadata_active$Drug),
  "Target" = SCOPEAnalysis::get_targets(metadata_active$Drug),
  "Line" = metadata_active$Line,
  row.names = rownames(profiles_active),
  stringsAsFactors = FALSE)

# Fix for slashes in targets
annotation$Target = gsub(
  pattern = "/", replacement = ", ", annotation$Target)

# Calculate the angles for each line
line_results = list()
enriched_drugs = list()
oddsratio_pathway = list()
oddsratio_target = list()
for(line in unique(substr(rownames(annotation), 1, 7))) {
  line_sel_indices = annotation$Line == line
  line_annotation = annotation[line_sel_indices, ]
  line_annotation$Line = NULL
  line_profiles = profiles_active[line_sel_indices, ]
  line_metadata = metadata_active[line_sel_indices, ]

  # Calc angles and cluster
  angles = get_angles(line_profiles)
  if(nrow(angles) < 2) next
  d = acos(angles)*180/pi
  hc = hclust(as.dist(d), method = "ward.D2")

  # Also cluster the principal components by correlation just for kicks
  cor_pca = cor(line_profiles)
  d_pca = as.dist((1 - cor_pca) / 2)
  hc_pca = hclust(d_pca, method = "ward.D2")

  # Annotate Pathway and Targets
  pathways = list()
  for(pathway in unique(line_annotation$Pathway)) {
    pathways[[pathway]] = grep(
      pattern = paste0("\\b", pathway, "\\b"),
      x = line_annotation$Pathway,
      ignore.case = TRUE)
  }
  # My cluster_enrichment function doesn't play well with NA values yet
```

Drug-Induced Phenotypes

```
pathways = pathways[!is.na(names(pathways))]
pathways_sig_vec = get_cluster_enrichment(
  clustering = hc, labels = pathways, min_cluster_size = min_cluster_size,
  max_annotated_labels = 7)
# 'Others' is too vague
pathways_sig_vec$Labels[pathways_sig_vec$Labels == "Others"] = NA

targets_split = tolower(line_annotation$Target)
targets = list()
for(target in unique(unlist(strsplit(targets_split, " ", )))) {
  targets[[target]] = grep(
    pattern = paste0("\b", target, "\b"),
    x = targets_split, ignore.case = TRUE)
}
# My cluster_enrichment function doesn't play well with NA values yet
targets = targets[!is.na(names(targets))]
targets_sig_vec = get_cluster_enrichment(
  clustering = hc, labels = targets, min_cluster_size = min_cluster_size,
  max_annotated_labels = 7)

line_annotation$Pathway = pathways_sig_vec$Labels
line_annotation$Target = targets_sig_vec$Labels

enriched_drugs[[line]] = data.frame(
  "Drug" = substr(rownames(line_annotation), 9, 100000L),
  "Line" = line,
  "Pathway" = annotation[annotation$Line == line, "Pathway"],
  "Target" = annotation[annotation$Line == line, "Target"],
  "Enriched.Target" = line_annotation$Target,
  "is.pathway.enriched" = !is.na(line_annotation$Pathway),
  "is.target.enriched" = !is.na(line_annotation$Target),
  stringsAsFactors = FALSE)

if(nrow(pathways_sig_vec$Fisher.Test) > 0) {
  oddsratio_pathway[[line]] = cbind.data.frame(
    "Line" = line, pathways_sig_vec$Fisher.Test)
}
if(nrow(targets_sig_vec$Fisher.Test) > 0) {
  oddsratio_target[[line]] = cbind.data.frame(
    "Line" = line, targets_sig_vec$Fisher.Test)
}

line_results[[line]] = list(
  "Annotation" = line_annotation,
  "Distances" = d,
  "Clustering" = hc,
  "PCAClustering" = hc_pca)
}

# Save enriched drugs
enriched_drugs = do.call(rbind, enriched_drugs)
```

Drug-Induced Phenotypes

```
rownames(enriched_drugs) = NULL
if(save_images) write.csv(
  x = enriched_drugs, file = "EnrichedDrugs.csv",
  quote = FALSE, row.names = FALSE)

# Save oddsratio Ratios
oddsratio_pathway = do.call(rbind, oddsratio_pathway)
rownames(oddsratio_pathway) = NULL
if(save_images) write.csv(
  x = oddsratio_pathway, file = "OddsRatio_Pathways.csv",
  quote = FALSE, row.names = FALSE)
oddsratio_target = do.call(rbind, oddsratio_target)
rownames(oddsratio_target) = NULL
if(save_images) write.csv(
  x = oddsratio_target, file = "OddsRatio_Target.csv",
  quote = FALSE, row.names = FALSE)
```

```
anno_colorScale = list(
  "Viability" = setNames(
    object = colorRampPalette(c("black", "white"))(
      length(unique(annotation$Viability))),
    nm = sort(unique(annotation$Viability)))))

# Plot all results
for(line in names(line_results)) {
  line_annotation = line_results[[line]]$Annotation
  line_profiles = profiles_active[annotation$Line == line,]
  line_color = anno_colorScale
  if(sum(!is.na(line_annotation$Pathway)) == 0) line_annotation$Pathway = NULL
  if(sum(!is.na(line_annotation$Target)) == 0) line_annotation$Target = NULL
  for(column in colnames(line_annotation)) {
    if(column %in% names(line_color)) next
    entries = na.omit(unique(line_annotation[[column]]))
    line_color[[column]] = setNames(
      object = unname(colorScale)[seq_along(entries)],
      nm = entries)
  }
  d = line_results[[line]]$Distances
  hc = line_results[[line]]$Clustering
  hc_pca = line_results[[line]]$PCAClustering

  # Heatmap of PCA components
  lp = line_profiles
  lp[lp > 0] = sqrt(lp[lp > 0])
  lp[lp < 0] = -sqrt(-lp[lp < 0])

  hm_range_limit = max(abs(lp))
  hm_breaks = seq(-hm_range_limit, hm_range_limit, length.out = 100)
  hm_legend_breaks = c(-floor(hm_range_limit), 0, floor(hm_range_limit))
  hm_legend_labels = hm_legend_breaks**2
  hm_legend_labels[hm_legend_breaks < 0] = -hm_legend_labels[hm_legend_breaks < 0]
```

Drug-Induced Phenotypes

```
hm_colorscale = colorRampPalette(
  c("#0082c8", "white", "#e6194b"))(length(hm_breaks))
pheatmap(
  t(lp), annotation_col = line_annotation,
  color = hm_colorscale, breaks = hm_breaks,
  legend_breaks = hm_legend_breaks, legend_labels = hm_legend_labels,
  annotation_colors = line_color, show_rownames = TRUE,
  show_colnames = FALSE, cluster_rows = hc_pca, cluster_cols = hc,
  main = sprintf("Angles Between Profile Vectors for %s", line),
  border_color = NA, cellheight = 7, cellwidth = 2)
if(save_images) pheatmap(
  t(lp), annotation_col = line_annotation,
  color = hm_colorscale, breaks = hm_breaks,
  legend_breaks = hm_legend_breaks, legend_labels = hm_legend_labels,
  annotation_colors = line_color, show_rownames = TRUE,
  show_colnames = FALSE, cluster_rows = hc_pca, cluster_cols = hc,
  main = sprintf("Angles Between Profile Vectors for %s", line),
  border_color = NA, cellheight = 7, cellwidth = 2, height = 10,
  filename = file.path(img_out_dir, sprintf("EffectVectorHeatmap_PCA_%s.pdf", line)))

# Symmetrical Heatmap (save as regular image and as super large image with rownames)
hm_colorscale = colorRampPalette(
  rev(c("#f7f7f7", "#f7f7f7", "#f7f7f7",
        "#E0F3F8", "#91BFDB", "#4575B4")))(150)
pheatmap(
  d, annotation_col = line_annotation,
  color = hm_colorscale, annotation_colors = line_color,
  show_rownames = FALSE, show_colnames = FALSE,
  cluster_rows = hc, cluster_cols = hc,
  main = sprintf("Angles Between Profile Vectors for %s", line),
  border_color = NA, cellheight = 1.25, cellwidth = 1.25)
if(save_images) pheatmap(
  d, annotation_col = line_annotation,
  color = hm_colorscale, annotation_colors = line_color,
  show_rownames = TRUE, show_colnames = FALSE,
  cluster_rows = hc, cluster_cols = hc,
  main = sprintf("Angles Between Profile Vectors for %s", line),
  # border_color = NA, cellheight = 1.25, cellwidth = 1.25, height = 10,
  border_color = NA, cellheight = 8, cellwidth = 8,
  filename = file.path(img_out_dir, sprintf("EffectVectorHeatmap_superlarge_%s.pdf", line)))
if(save_images) pheatmap(
  d, annotation_col = line_annotation,
  color = hm_colorscale, annotation_colors = line_color,
  show_rownames = FALSE, show_colnames = FALSE,
  cluster_rows = hc, cluster_cols = hc,
  main = sprintf("Angles Between Profile Vectors for %s", line),
  border_color = NA, cellheight = 1.25, cellwidth = 1.25, height = 10,
  filename = file.path(img_out_dir, sprintf("EffectVectorHeatmap_%s.pdf", line)))

# Force-directed graph of active drugs
# Set up adjacency matrix
```

Drug-Induced Phenotypes

```
adjmatrix = d
adjmatrix[adjmatrix > 45] = 0

# Resort matrix to make sure annotated nodes are above unannotated ones
sort_order = order(line_annotation$Target, na.last = FALSE)
adjmatrix = adjmatrix[sort_order, sort_order]

# Remove drugs without any connections
sel_indices = colSums(adjmatrix != 0) != 0
adjmatrix = adjmatrix[sel_indices, sel_indices]

# Generate network
adjmatrix = Matrix::Matrix(data = adjmatrix, sparse = TRUE)
net = graph_from_adjacency_matrix(
  adjmatrix, mode = "undirected", diag = FALSE, weighted = TRUE)

# Assign targets and colors
V(net)$target = line_annotation[match(
  V(net)$name, rownames(line_annotation)), "Target"]
V(net)$color = V(net)$target
for(target in na.omit(unique(V(net)$target))) {
  regex = paste0("\\b", target, "\\b")
  V(net)$color = gsub(regex, line_color$Target[target], V(net)$color)}
V(net)$color[is.na(V(net)$color)] = "white"

# Assign size
V(net)$size = 5
V(net)[!is.na(V(net)$target)]$size = 7.5

# Create plot
e = get.edgelist(net, names = FALSE)
l = qgraph.layout.fruchtermanreingold(
  edgelist = e, vcount = vcount(net),
  area=20*(vcount(net)^2), repulse.rad=(vcount(net)^3.1))
# l = layout_with_graphopt(graph = net, niter = 1000, mass = 100)
plot.igraph(net, vertex.label = NA, layout=l,
            edge.width = 0.5, edge.color = "lightgray",
            main = line)
if(save_images) {
  pdf(file = file.path(img_out_dir, sprintf("EffectVectorNetwork_%s.pdf", line)),
       width = 7, height = 7, useDingbats = FALSE)
  plot.igraph(net, vertex.label = NA, layout=l,
              edge.width = 0.5, edge.color = "lightgray",
              main = line)
  dev.off()}

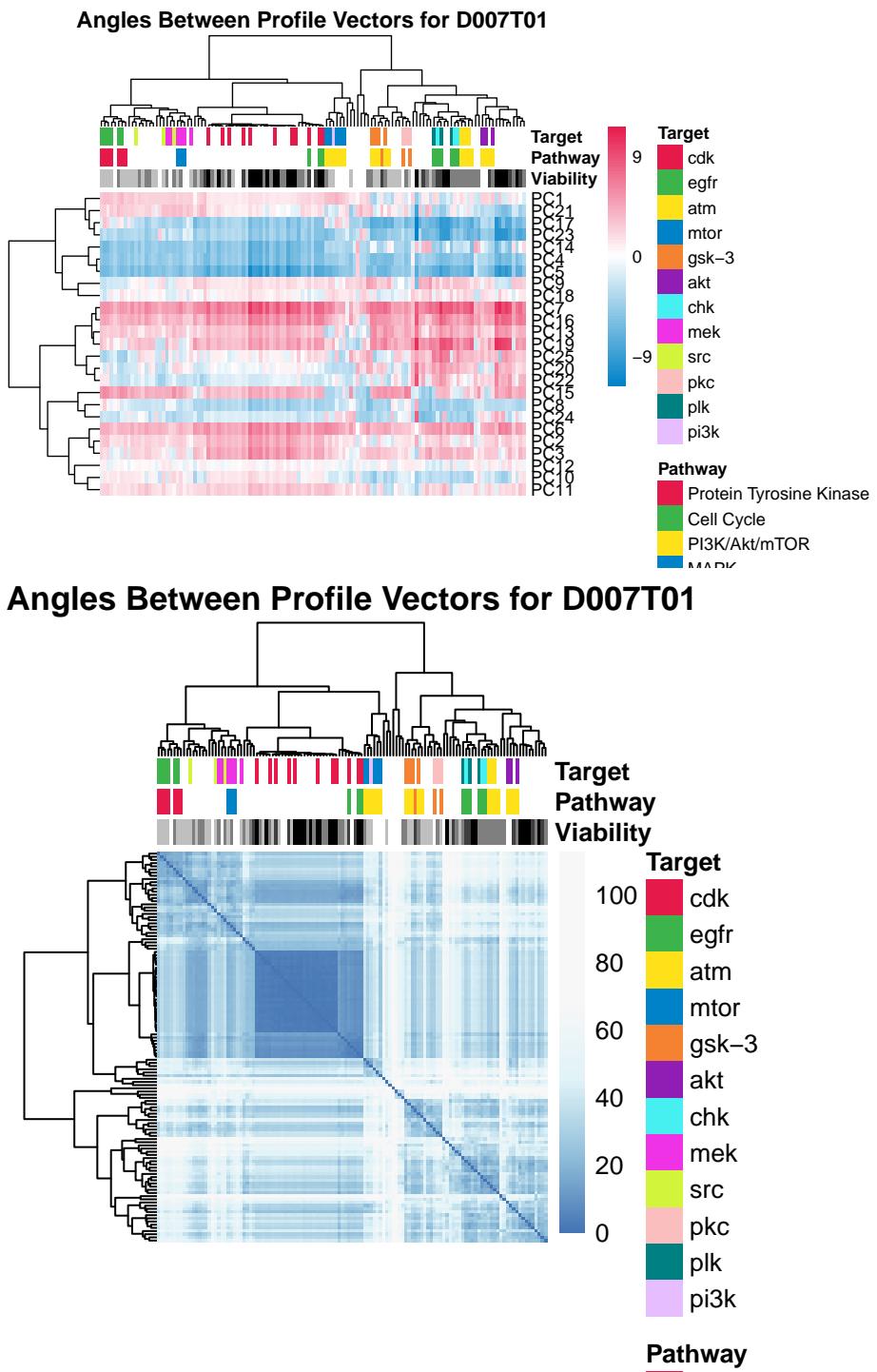
# # t-SNE of full profiles
# line_profiles_full = profiles[metadata$Line == line, ]
# tsne = Rtsne(X = line_profiles_full, perplexity = 50, max_iter = 1000)
# tsnedf = data.frame(
#   "X" = tsne$Y[,1],
```

Drug-Induced Phenotypes

```
# "Y" = tsne$Y[,2],
# "isActive" = rownames(line_profiles_full) %in% rownames(line_profiles),
# "Viability" = line_annotation[rownames(line_profiles_full), "Viability"],
# "Pathway" = line_annotation[rownames(line_profiles_full), "Pathway"],
# "Target" = line_annotation[rownames(line_profiles_full), "Target"],
# row.names = rownames(line_profiles_full)
# )
# ggplot() +
#   # Inactive drugs
#   geom_point(data = tsnedf[!tsnedf$isActive, ],
#             mapping = aes(x = X, y = Y),
#             color = "lightgray") +
#   # Active Drugs
#   geom_point(data = tsnedf[tsnedf$isActive, ],
#             mapping = aes(x = X, y = Y)) +
#   scale_color_manual(values = line_color$Target) +
#   # Enriched Drugs (Target)
#   geom_point(data = tsnedf[!is.na(tsnedf$Target), ],
#             mapping = aes(
#               x = X, y = Y,
#               color = tsnedf[!is.na(tsnedf$Target), "Target"])) +
#   theme_vignette() + ggtitle(paste0("t-SNE of all drugs ", line)) +
#   labs(color = "Target")
# if(save_images) ggsave(
#   filename = file.path(img_out_dir, sprintf(
#     "EffectVector_allDrugs_tSNE_%s.pdf", line)),
#   width = 8, height = 8, useDingbats = FALSE)
#
# # t-SNE of active drugs
# tsne = Rtsne(X = line_profiles, perplexity = 20, max_iter = 1000)
# tsnedf = data.frame(
#   "X" = tsne$Y[,1],
#   "Y" = tsne$Y[,2],
#   "Viability" = line_annotation[rownames(line_profiles), "Viability"],
#   "Pathway" = line_annotation[rownames(line_profiles), "Pathway"],
#   "Target" = line_annotation[rownames(line_profiles), "Target"],
#   row.names = rownames(line_profiles)
# )
# ggplot() +
#   geom_point(data = tsnedf, mapping = aes(x = X, y = Y)) +
#   scale_color_manual(values = line_color$Target) +
#   # Enriched Drugs (Target)
#   geom_point(data = tsnedf[!is.na(tsnedf$Target), ],
#             mapping = aes(
#               x = X, y = Y,
#               color = tsnedf[!is.na(tsnedf$Target), "Target"])) +
#   theme_vignette() + ggtitle(paste0("t-SNE of active drugs ", line)) +
#   labs(color = "Target")
# if(save_images) ggsave(
#   filename = file.path(img_out_dir, sprintf(
#     "EffectVector_activeDrugs_tSNE_%s.pdf", line)),
```

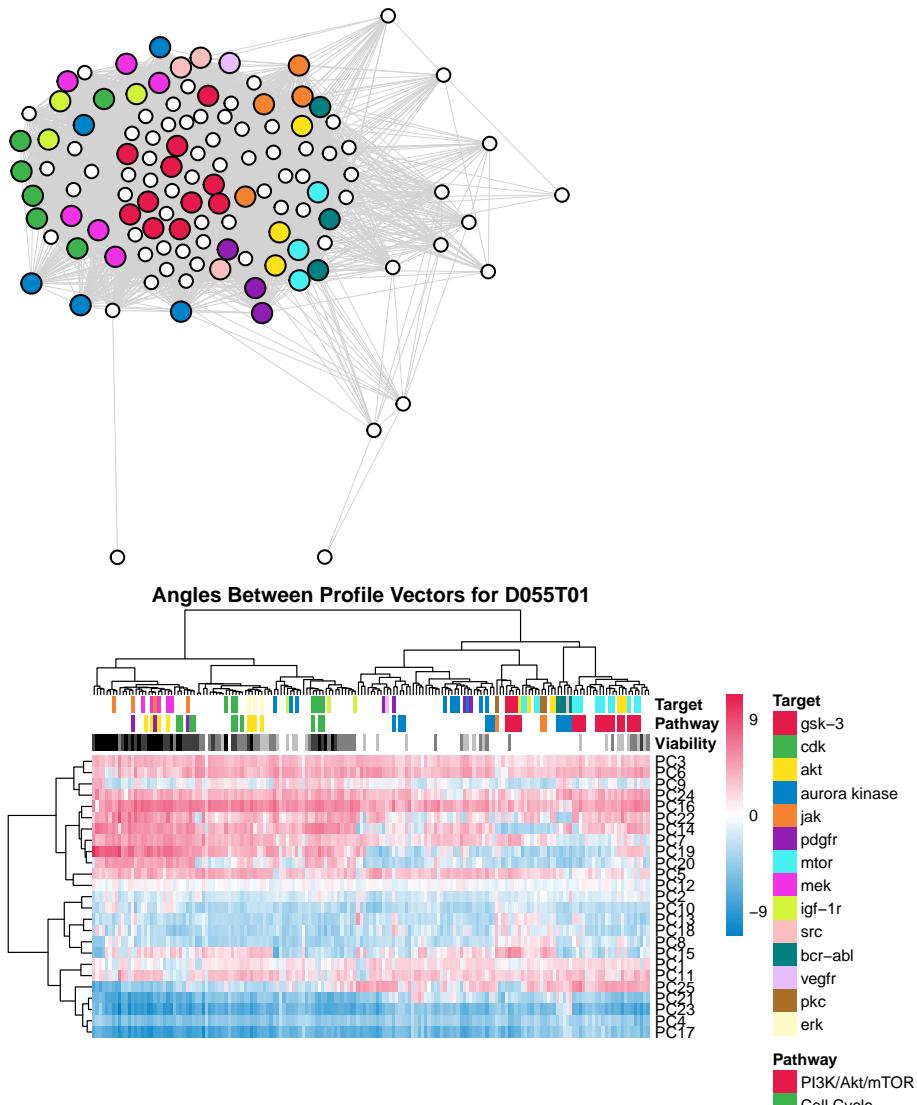
Drug-Induced Phenotypes

```
#   width = 8, height = 8, useDingbats = FALSE)  
}
```

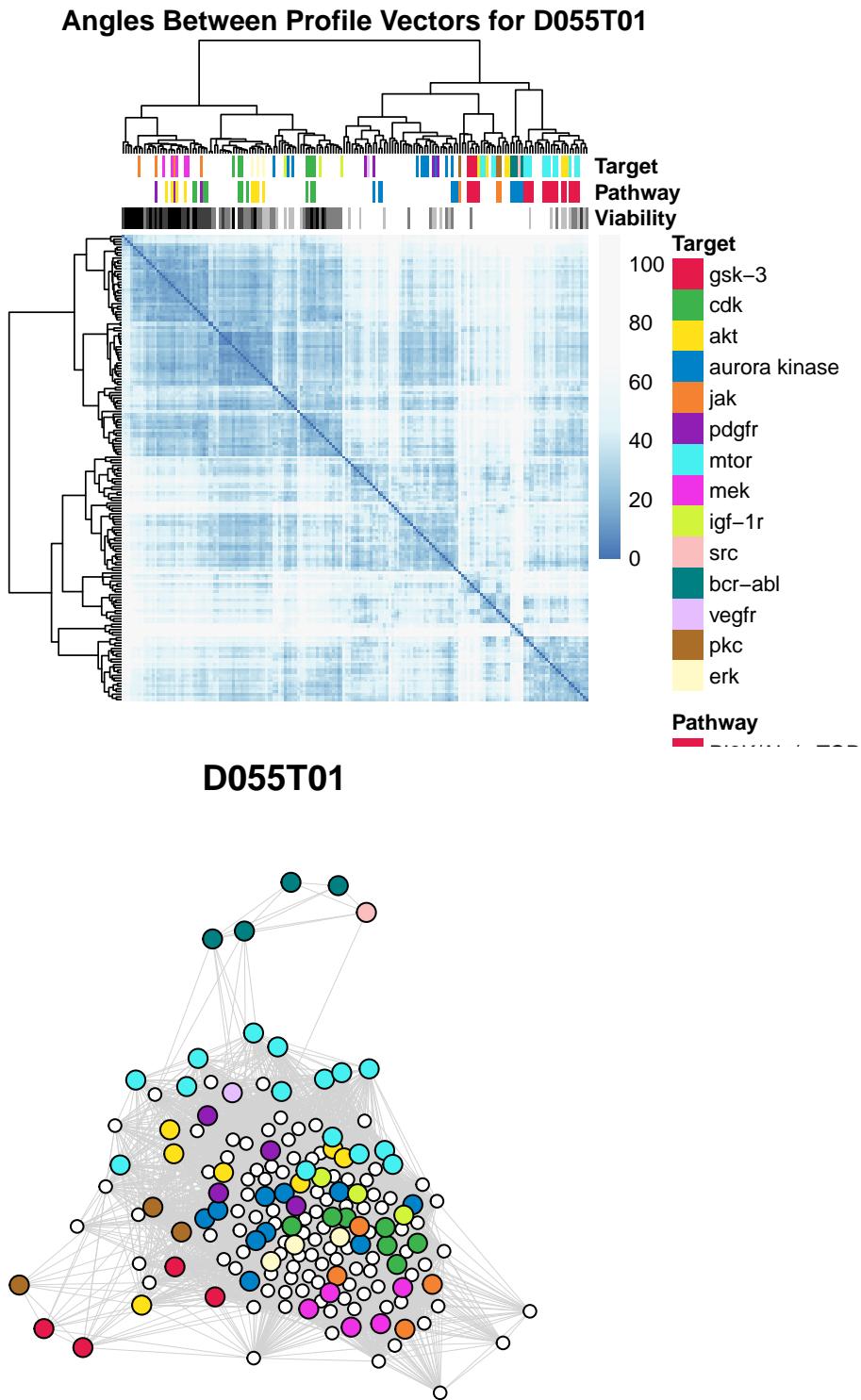


Drug-Induced Phenotypes

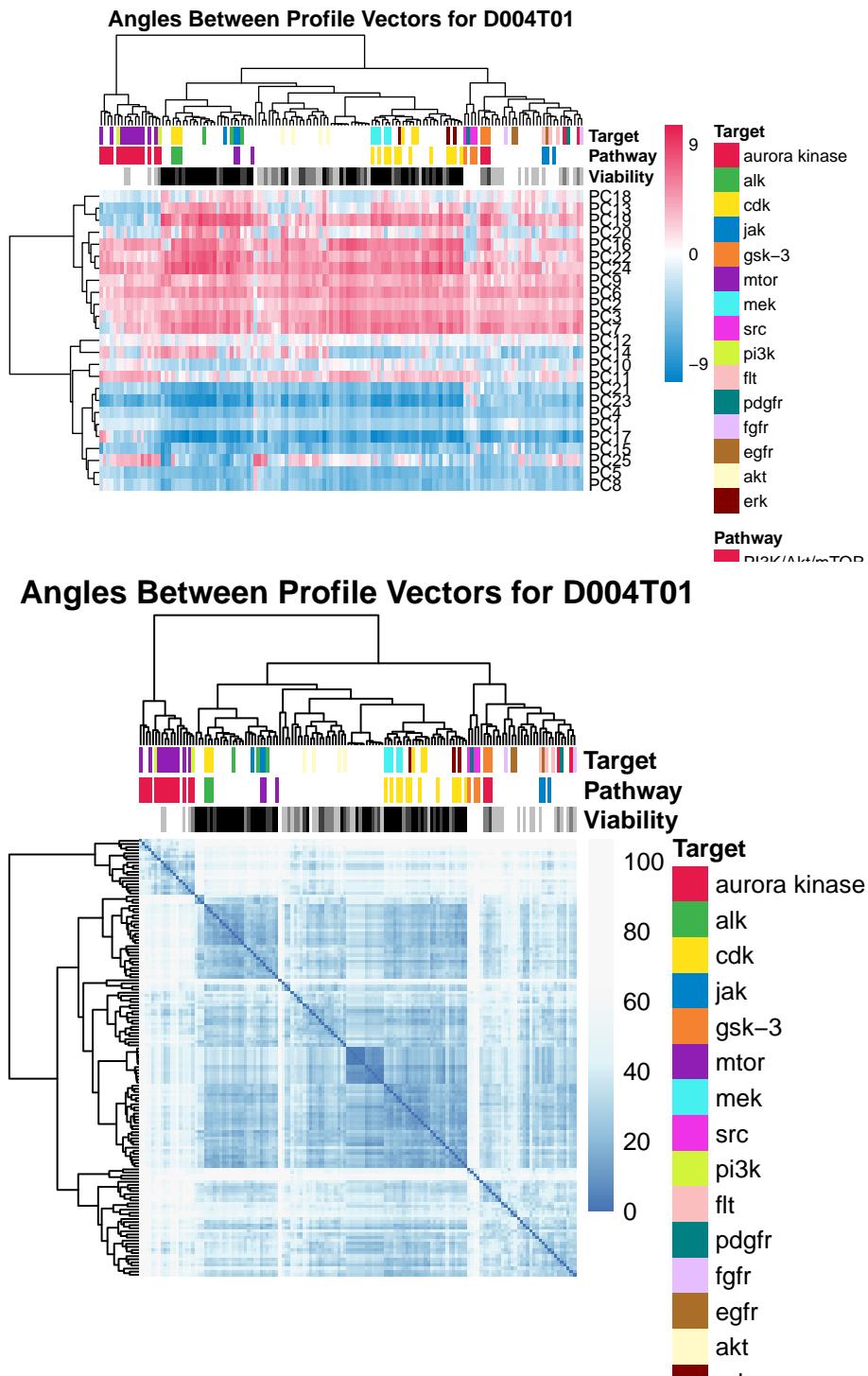
D007T01



Drug-Induced Phenotypes

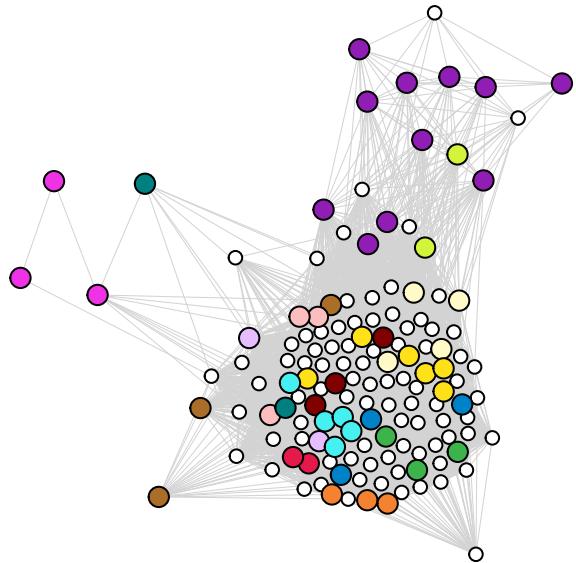


Drug-Induced Phenotypes

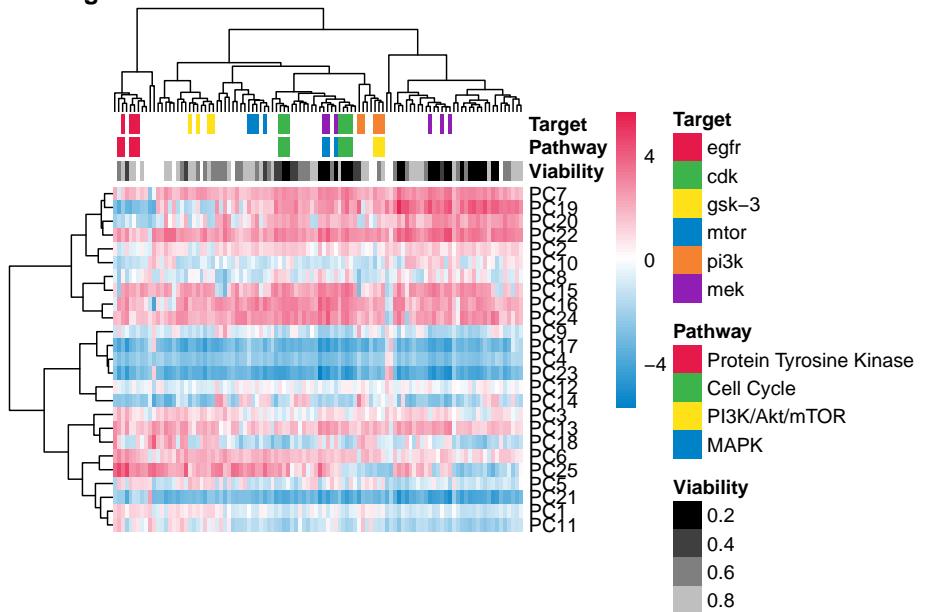


Drug-Induced Phenotypes

D004T01

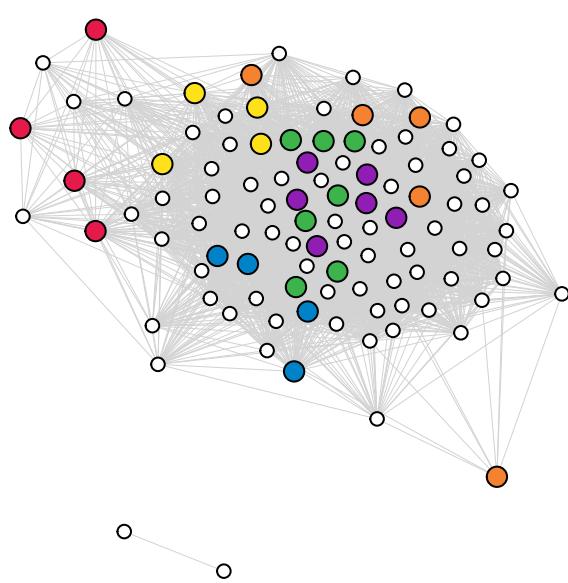
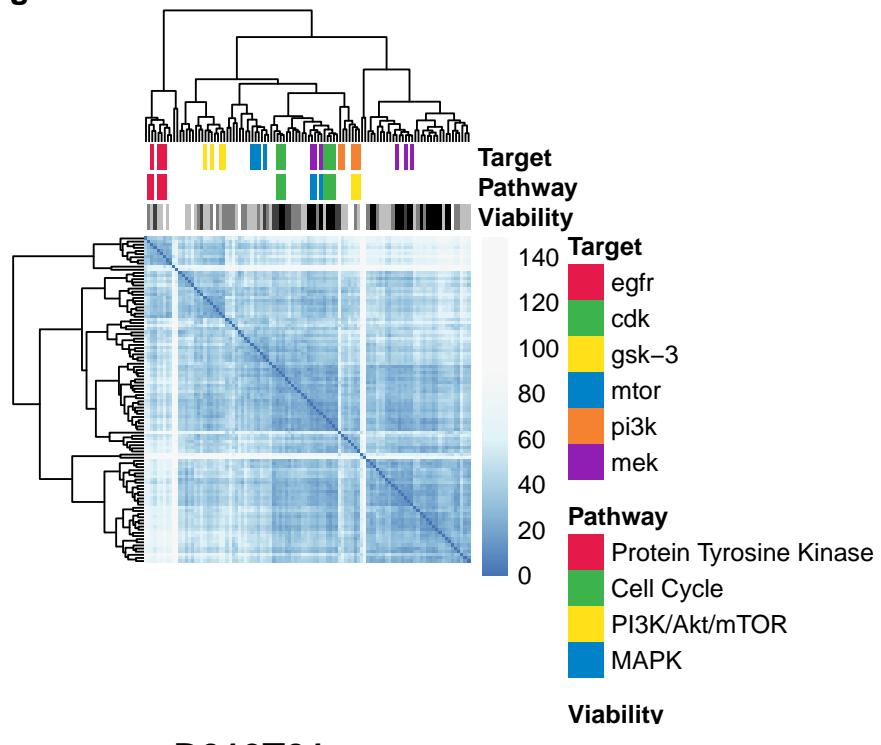


Angles Between Profile Vectors for D010T01

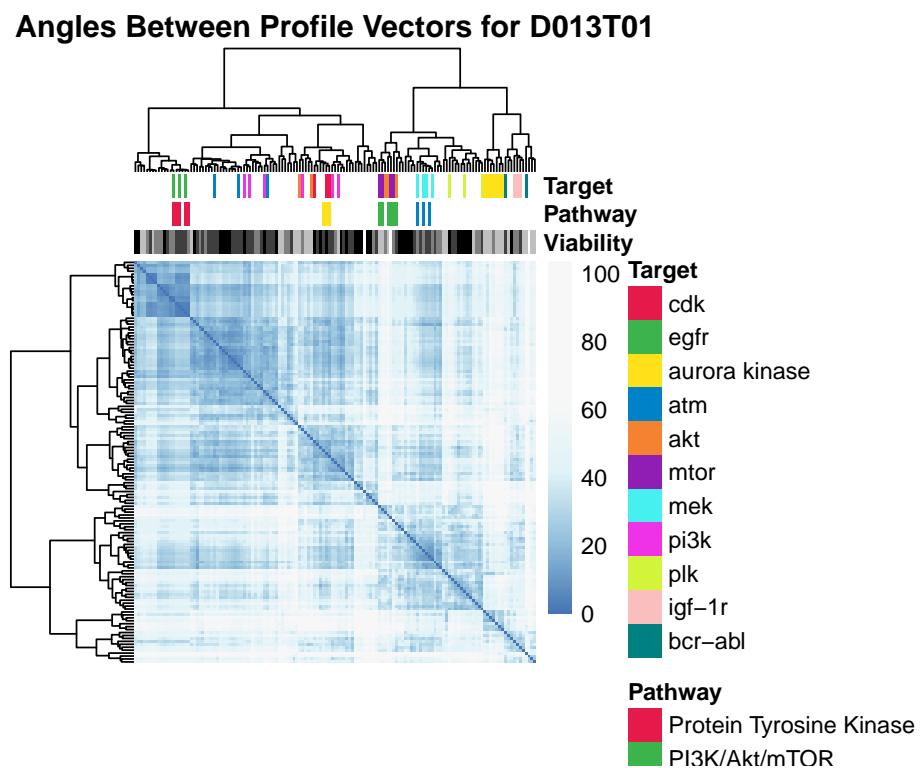
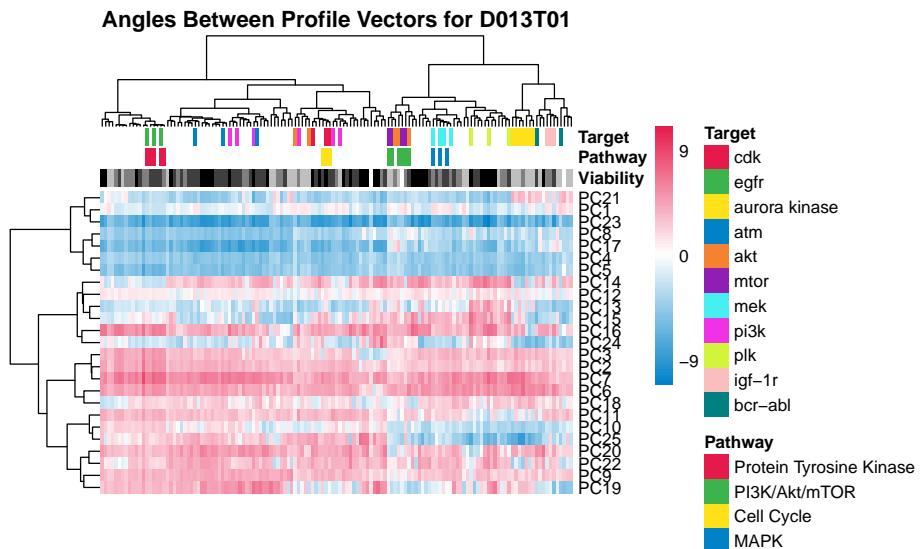


Drug-Induced Phenotypes

Angles Between Profile Vectors for D010T01

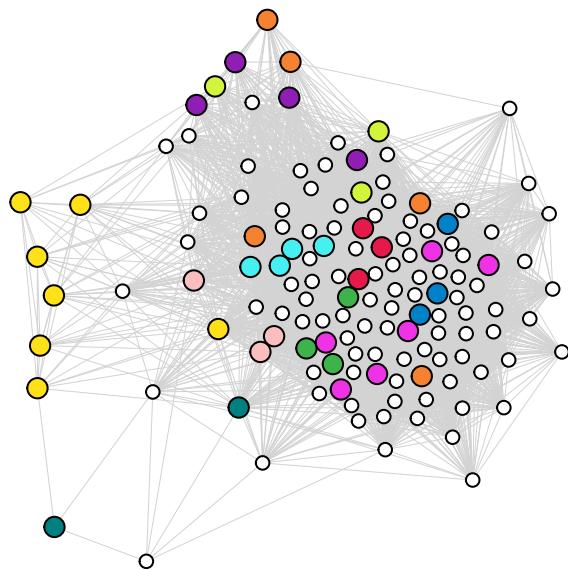


Drug-Induced Phenotypes

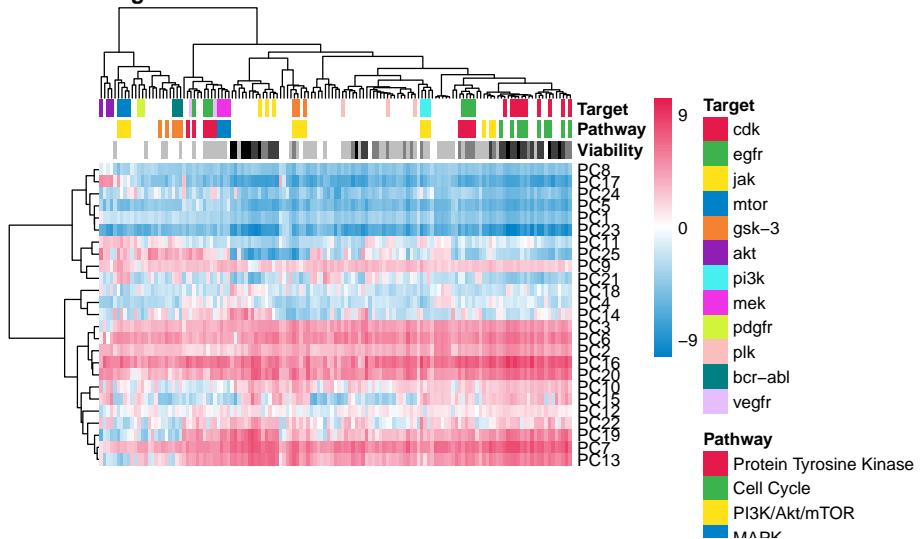


Drug-Induced Phenotypes

D013T01

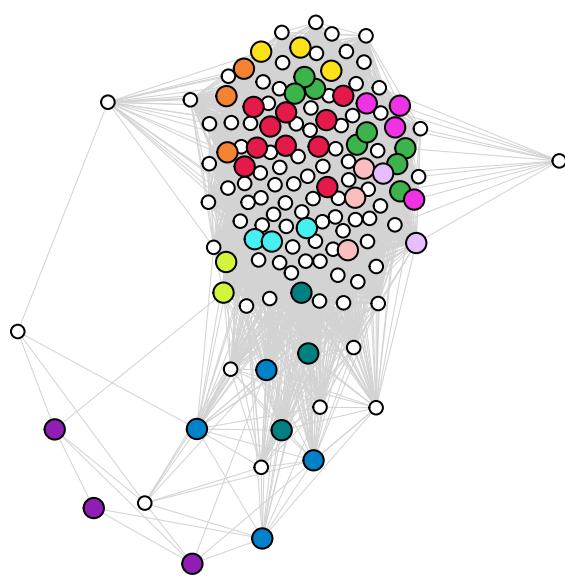
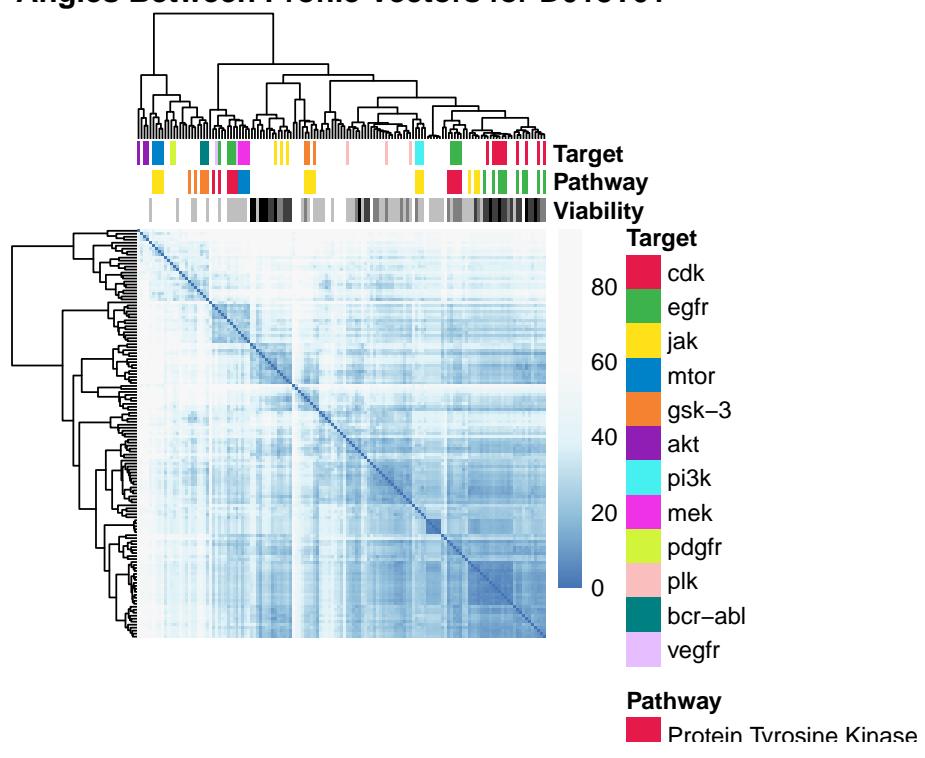


Angles Between Profile Vectors for D018T01

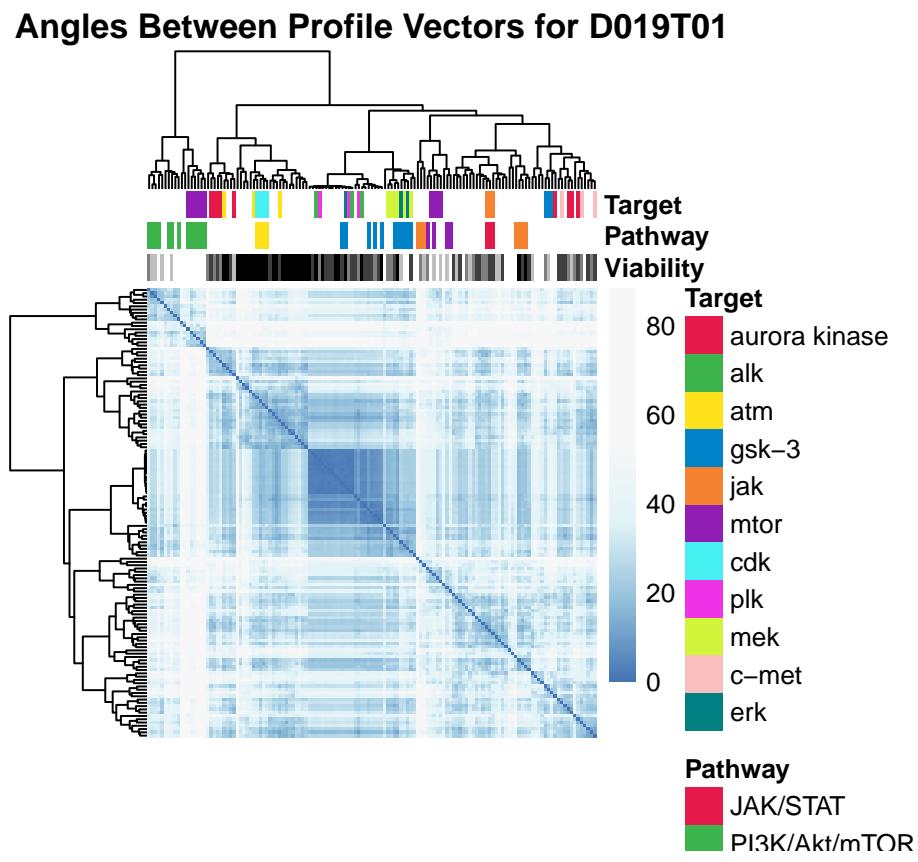
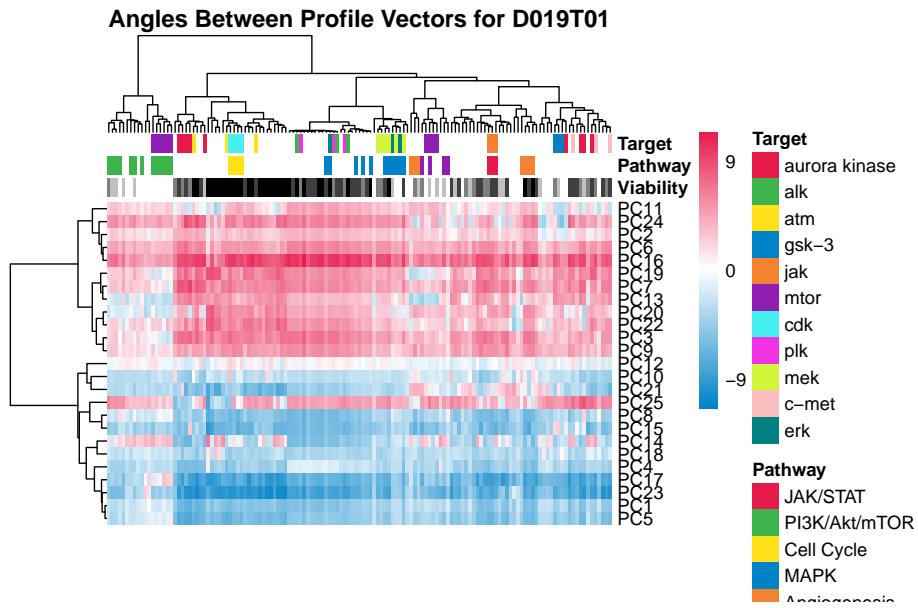


Drug-Induced Phenotypes

Angles Between Profile Vectors for D018T01

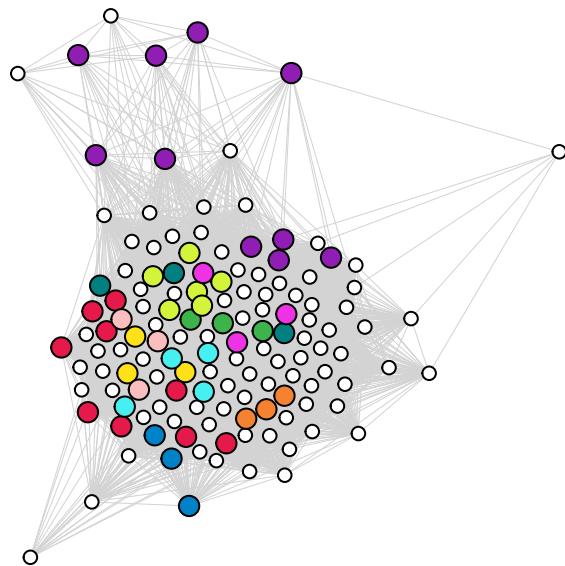


Drug-Induced Phenotypes

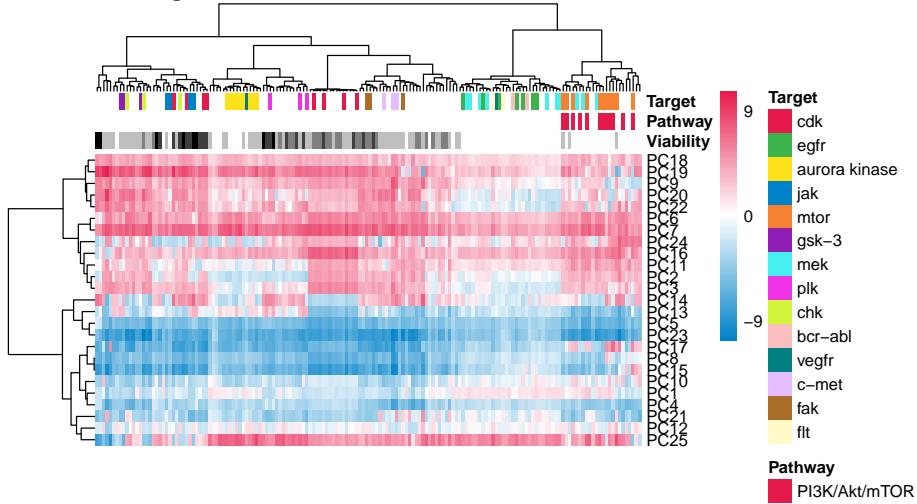


Drug-Induced Phenotypes

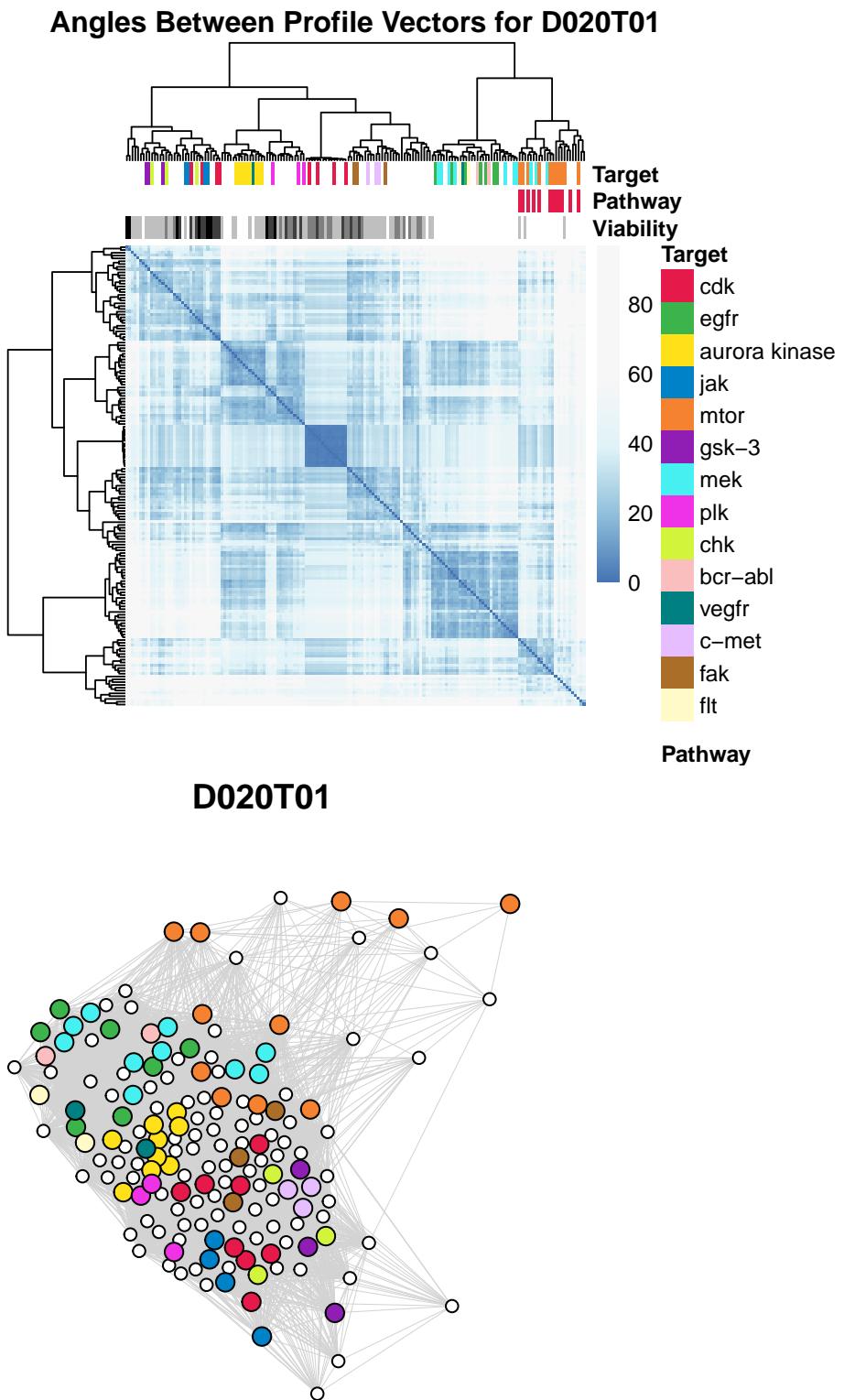
D019T01



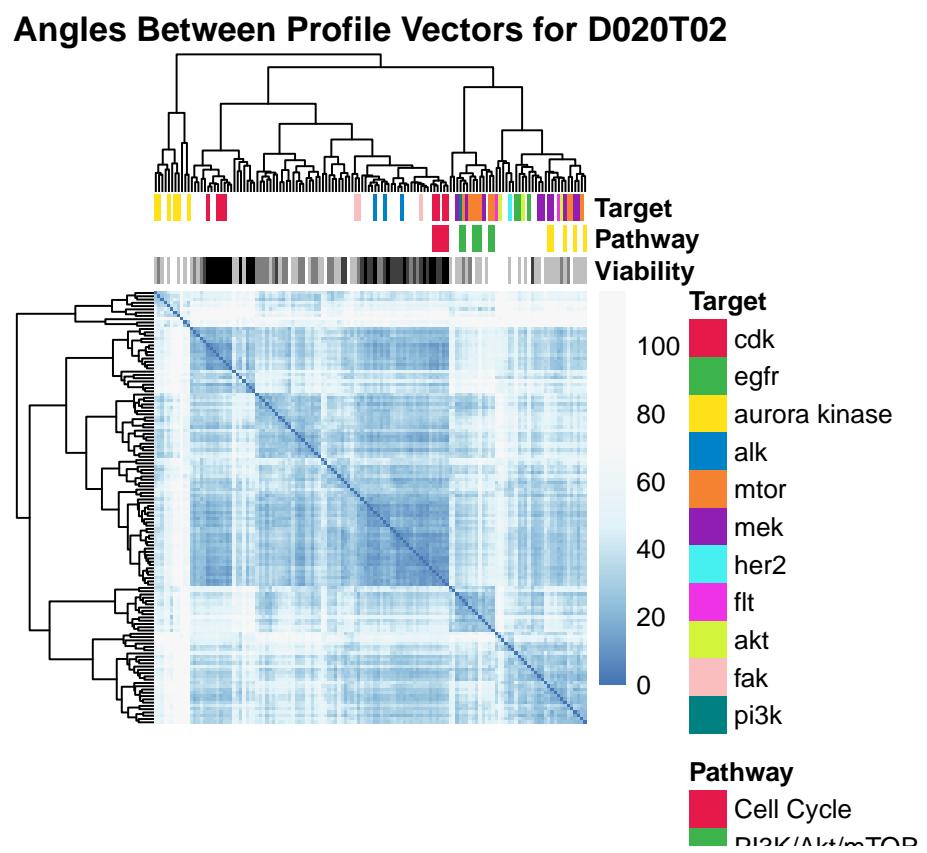
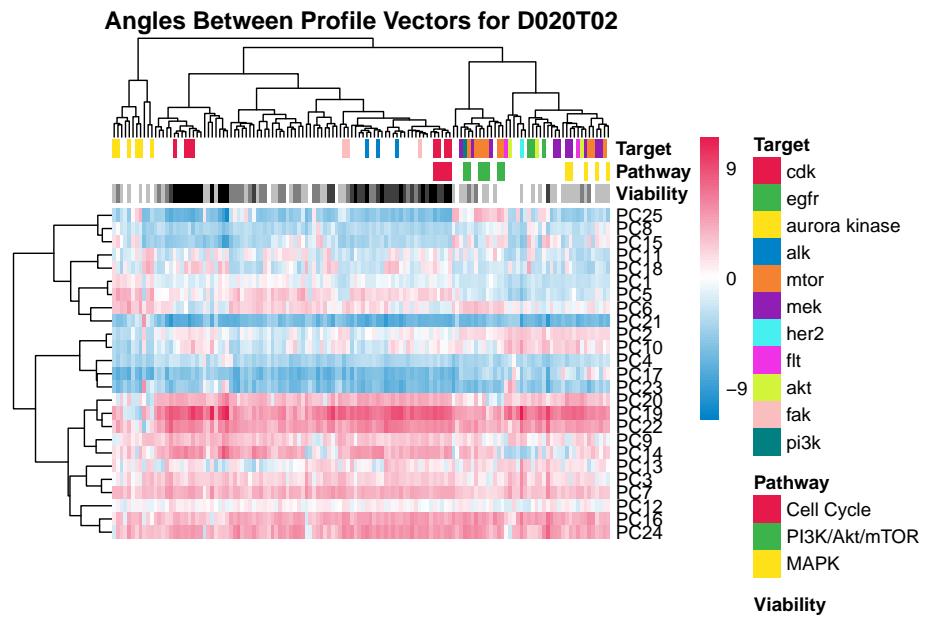
Angles Between Profile Vectors for D020T01



Drug-Induced Phenotypes

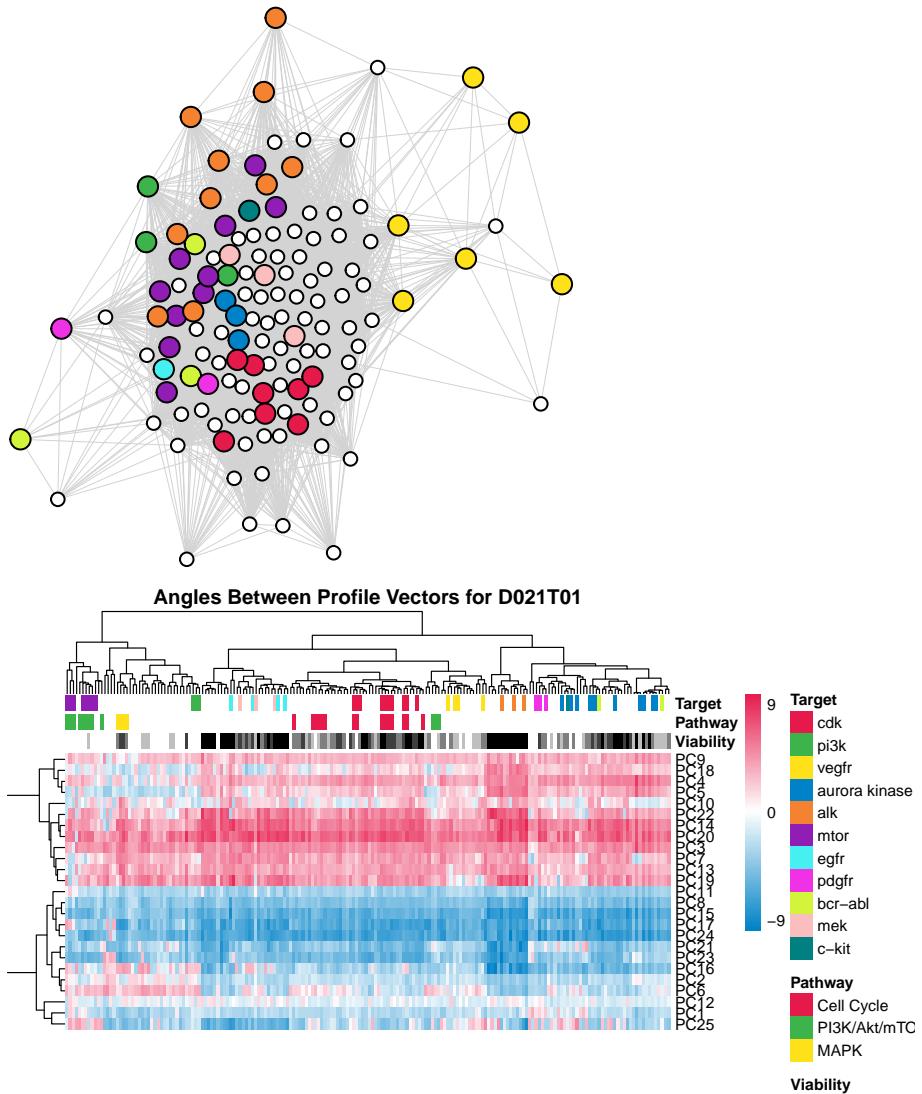


Drug-Induced Phenotypes

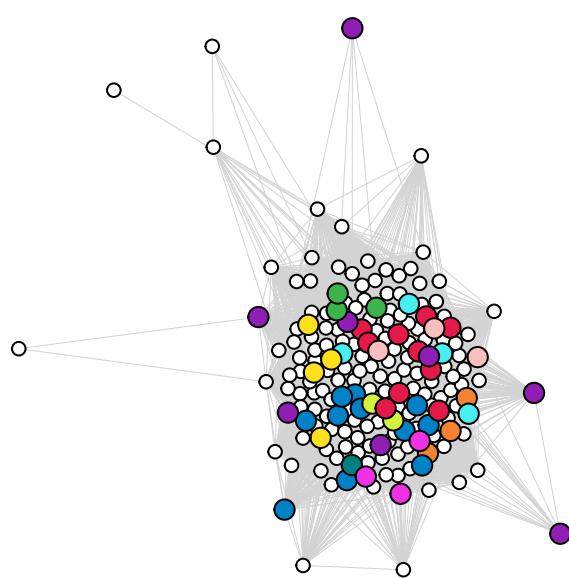
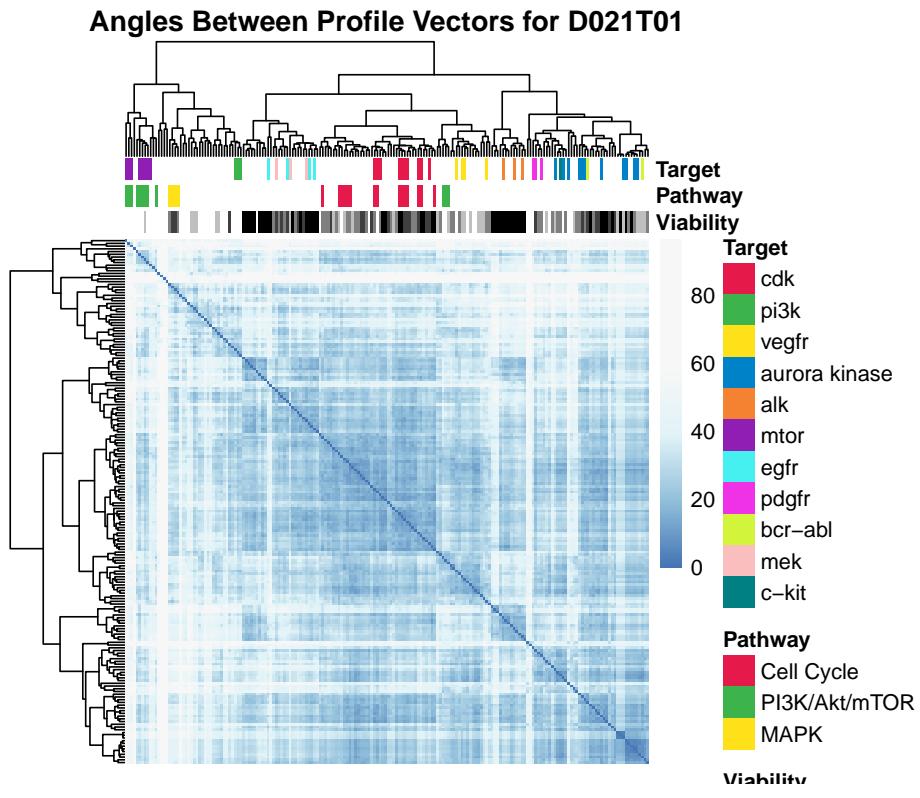


Drug-Induced Phenotypes

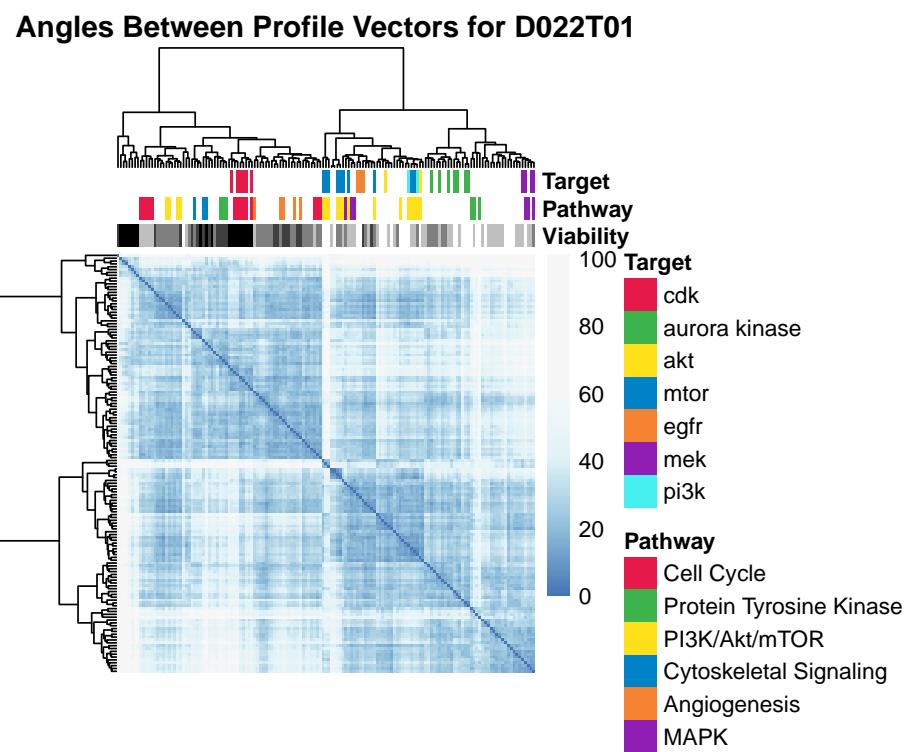
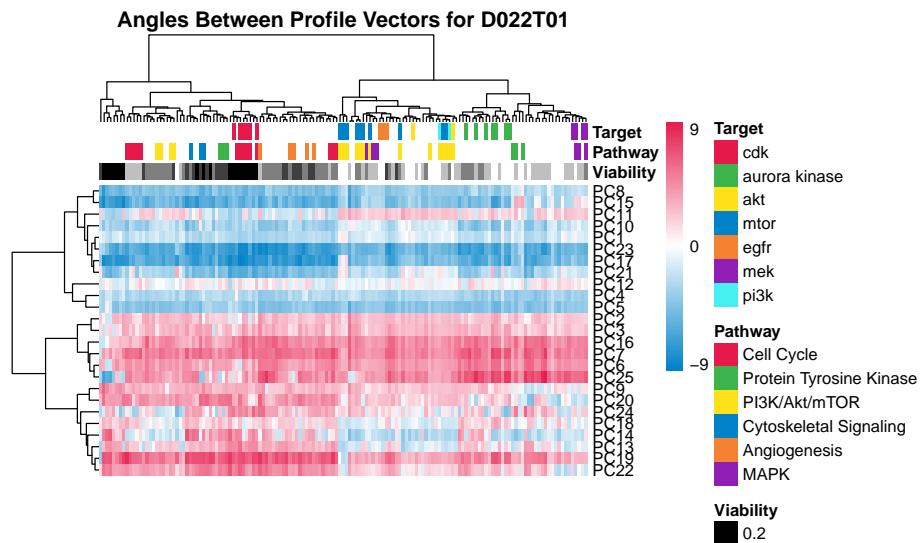
D020T02



Drug-Induced Phenotypes

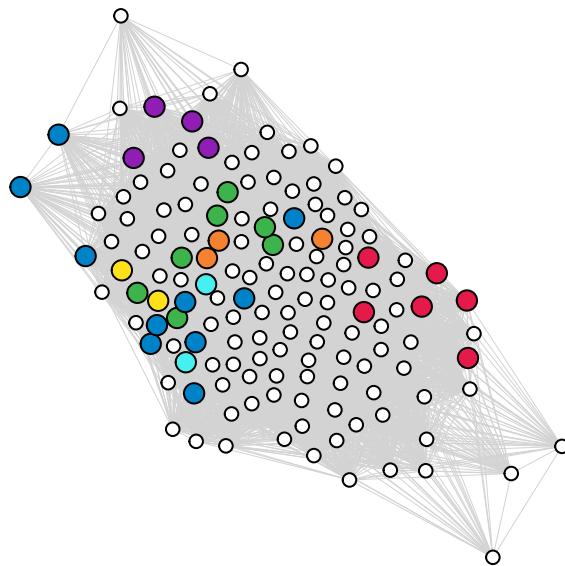


Drug-Induced Phenotypes

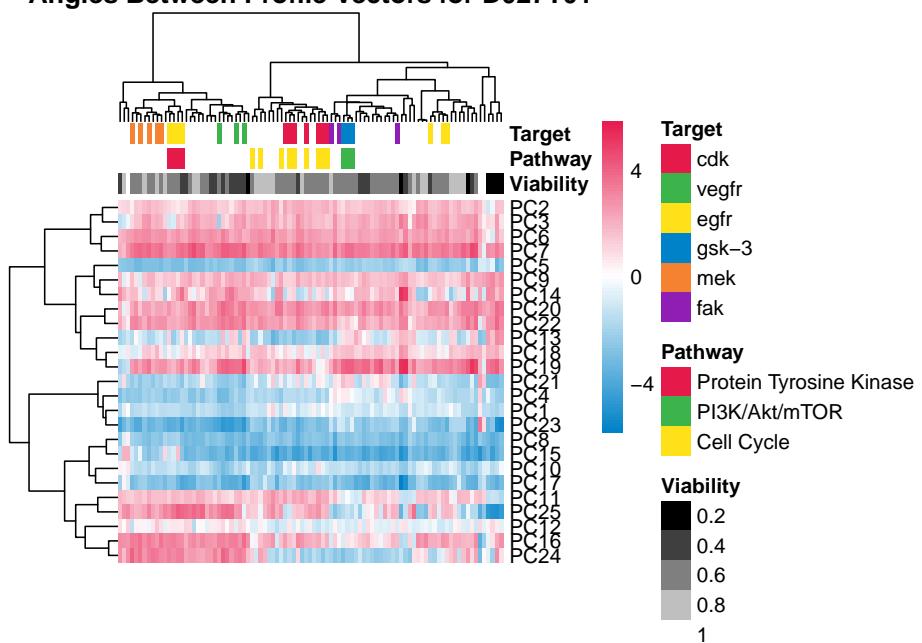


Drug-Induced Phenotypes

D022T01

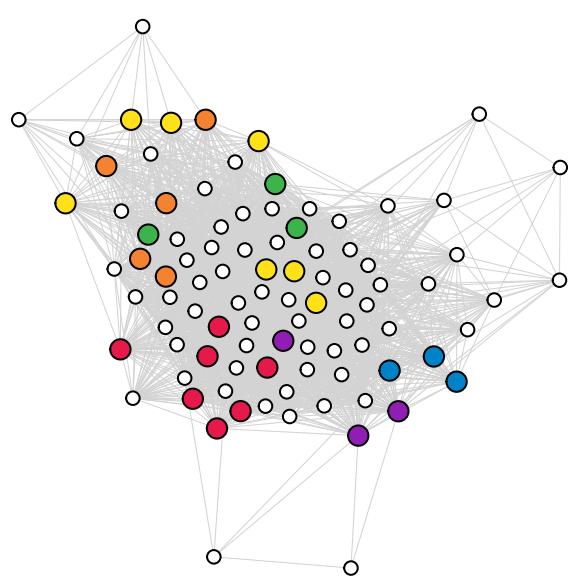
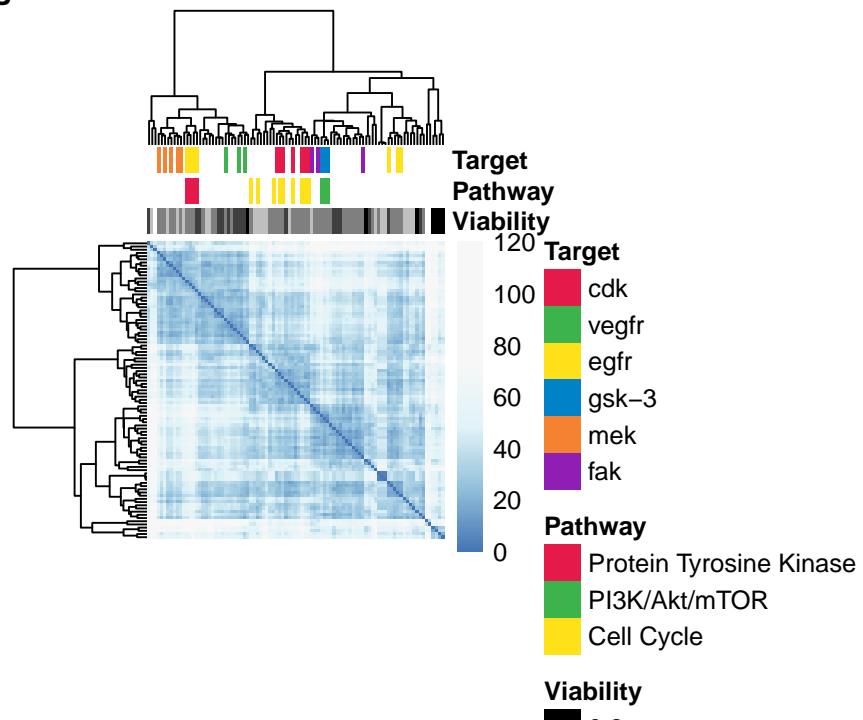


Angles Between Profile Vectors for D027T01

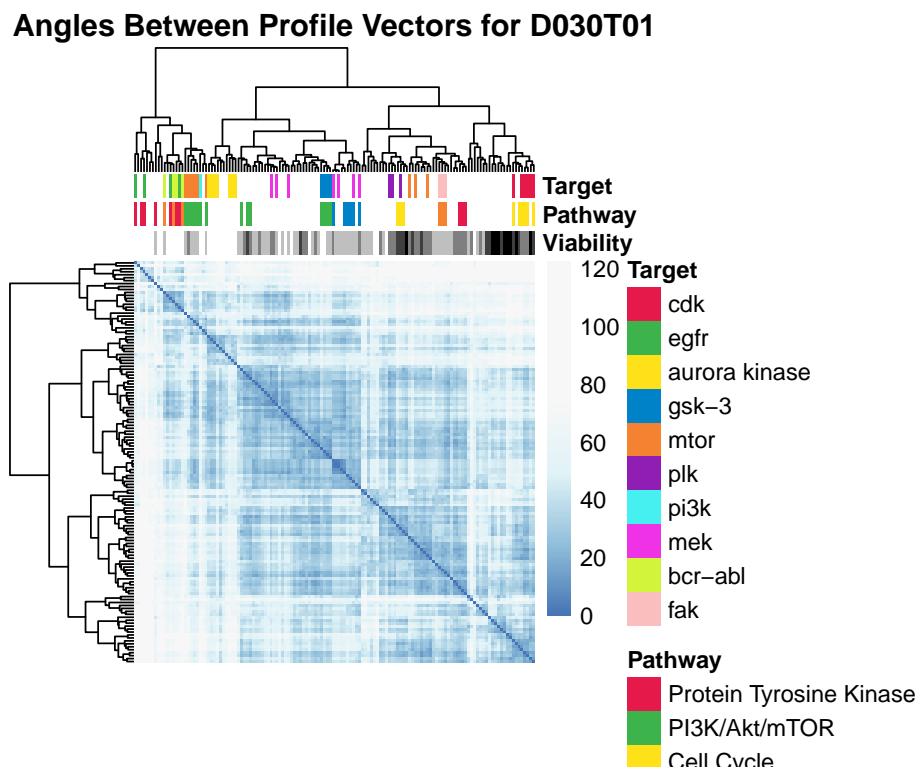
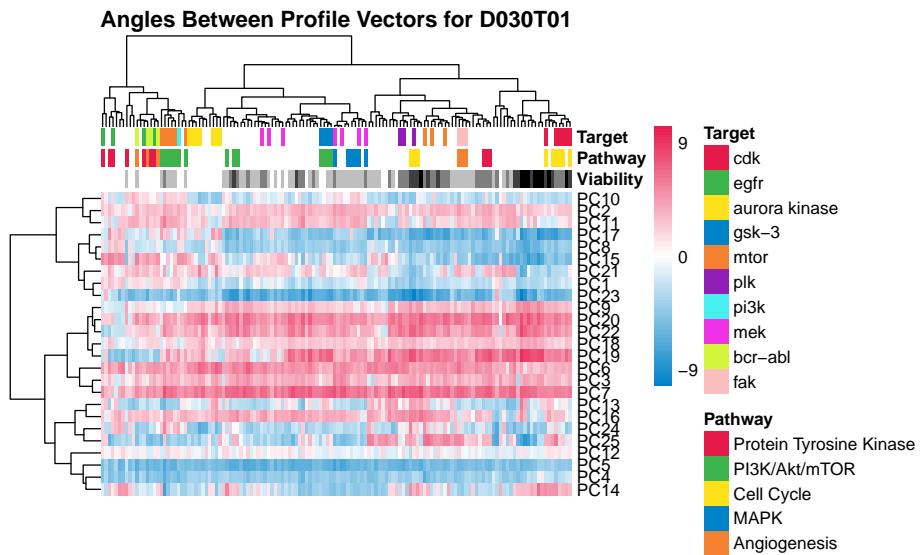


Drug-Induced Phenotypes

Angles Between Profile Vectors for D027T01

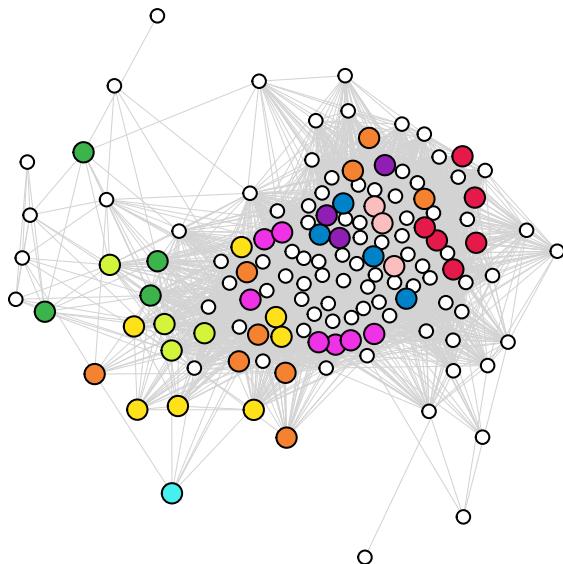


Drug-Induced Phenotypes

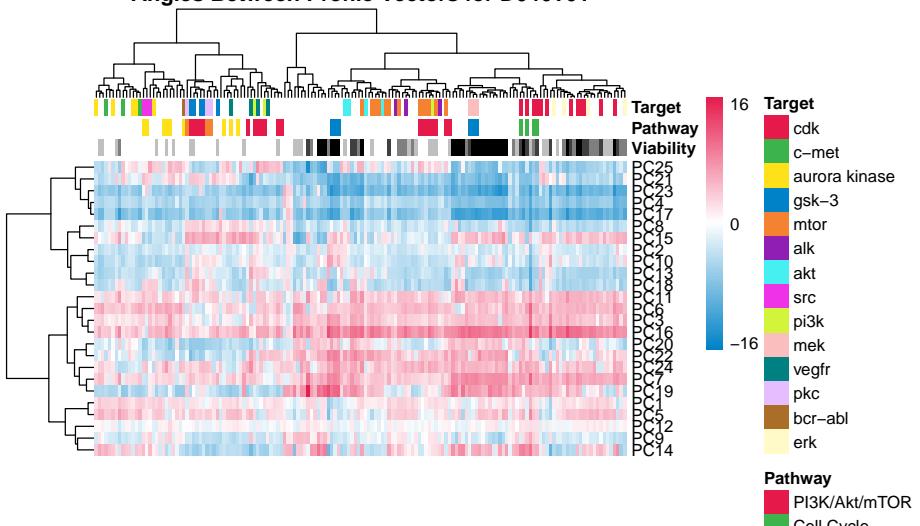


Drug-Induced Phenotypes

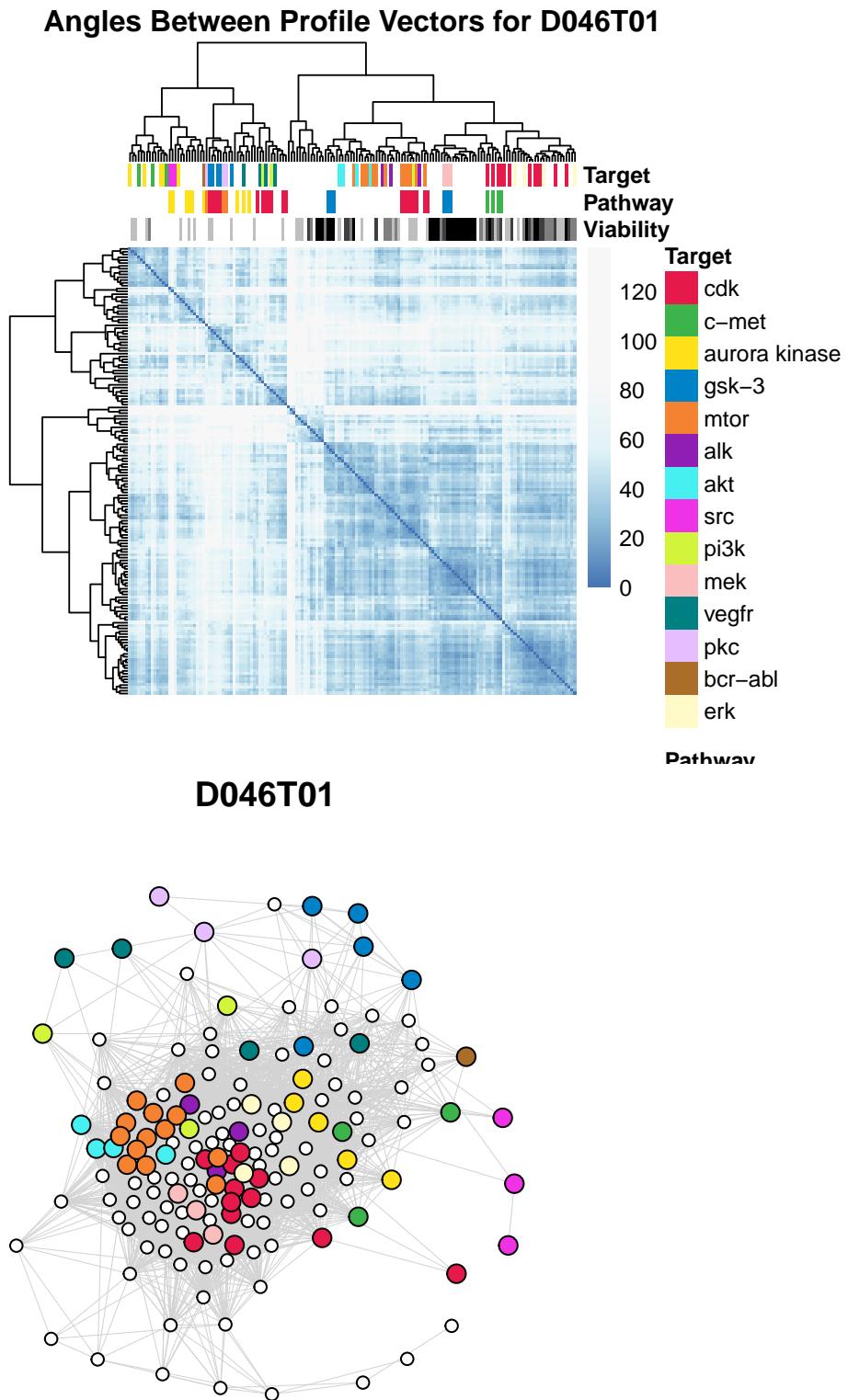
D030T01



Angles Between Profile Vectors for D046T01



Drug-Induced Phenotypes



Drug-Induced Phenotypes

3.1.1 Closer Look at D004T01

D004T01 has a notable non-lethal cluster associated with mTOR

```
line = "D004T01"
line_annotation = line_results[[line]]$Annotation
line_profiles = profiles_active[annotation$Line == line,]
line_color = anno_colorScale
if(sum(!is.na(line_annotation$Pathway)) == 0) line_annotation$Pathway = NULL
if(sum(!is.na(line_annotation$Target)) == 0) line_annotation$Target = NULL
for(column in colnames(line_annotation)) {
  if(column %in% names(line_color)) next
  entries = na.omit(unique(line_annotation[[column]]))
  line_color[[column]] = setNames(
    object = unname(colorScale)[seq_along(entries)],
    nm = entries)
}
d = line_results[[line]]$Distances
hc = line_results[[line]]$Clustering
hc_pca = line_results[[line]]$PCAClustering

labels = cutree(tree = hc, k = 2)
mtor_label = unique(labels[which(line_annotation$Target == "mtor")])
if(length(mtor_label) != 1) stop(
  "Critical Error with mTOR label for D004T01")
mtor_drugs = names(which(labels == mtor_label))

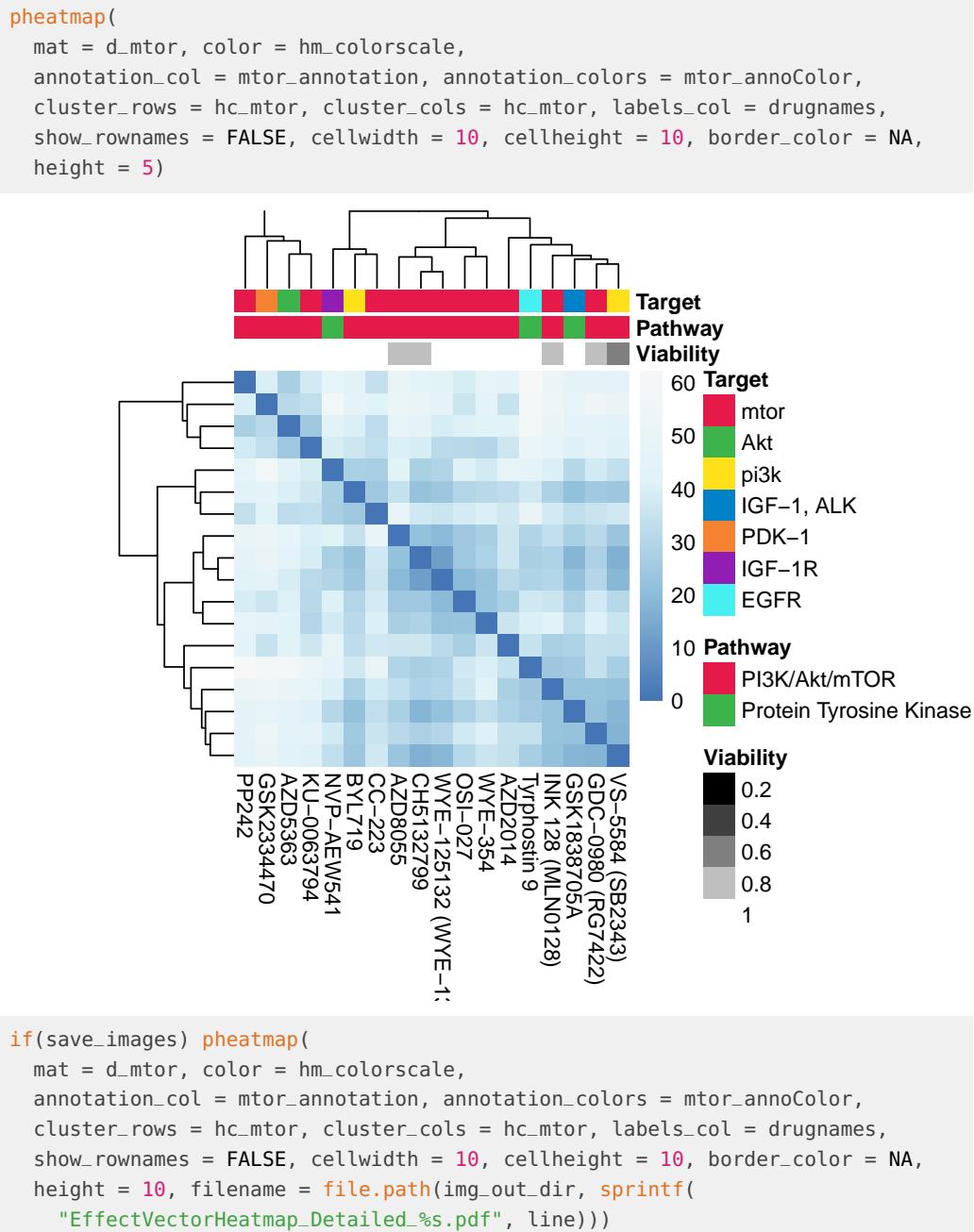
mtor_annotation = annotation[mtor_drugs, ]
mtor_annotation$Target = ifelse(
  test = !is.na(line_annotation[mtor_drugs, "Target"]),
  yes = line_annotation[mtor_drugs, "Target"],
  no = mtor_annotation[mtor_drugs, "Target"])
mtor_annotation$Line = NULL
mtor_annoColor = anno_colorScale
if(sum(!is.na(mtor_annotation$Pathway)) == 0) mtor_annotation$Pathway = NULL
if(sum(!is.na(mtor_annotation$Target)) == 0) mtor_annotation$Target = NULL
for(column in colnames(mtor_annotation)) {
  if(column %in% names(mtor_annoColor)) next
  entries = na.omit(unique(mtor_annotation[[column]]))
  mtor_annoColor[[column]] = setNames(
    object = unname(colorScale)[seq_along(entries)],
    nm = entries)
}

d_mtor = d[labels == mtor_label, labels == mtor_label]
hc_mtor = hclust(as.dist(d_mtor), "ward.D2")

hm_colorscale = colorRampPalette(
  rev(c("#f7f7f7", "#E0F3F8", "#91BFDB", "#4575B4")))(150)

drugnames = substr(mtor_drugs, 9, 100)
```

Drug-Induced Phenotypes



3.2 Enrichment Heatmap

I look at enrichment heatmaps, i.e. which cell lines have an enrichment in which pathways / targets.

Drug-Induced Phenotypes

3.2.1 Log-Odds Heatmap

I create a heatmap of the enrichment (log odds ratio) to determine which pathways and targets are enriched in each cell line

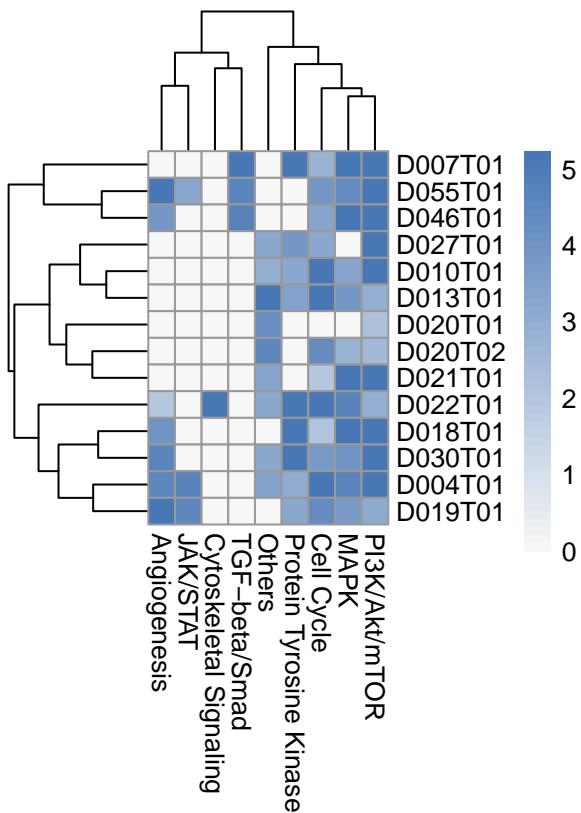
```
heatmap_pathways = oddsratio_pathway[, c("Line", "Label", "OddsRatio")]
heatmap_pathways = acast(
  data = heatmap_pathways,
  formula = Line ~ Label, value.var = "OddsRatio",
  fun.aggregate = max, fill = 0)
# Replace Inf with the finite maximum
heatmap_pathways[is.infinite(heatmap_pathways)] = max(
  heatmap_pathways[!is.infinite(heatmap_pathways)])
heatmap_pathways[heatmap_pathways != 0] = log2(
  heatmap_pathways[heatmap_pathways != 0])
hc_pathways = as.dist((1 - cor(heatmap_pathways)) / 2)
hc_pathways_line = as.dist((1 - cor(t(heatmap_pathways))) / 2)

heatmap_targets = oddsratio_target[, c("Line", "Label", "OddsRatio")]
heatmap_targets = acast(
  data = heatmap_targets,
  formula = Line ~ Label, value.var = "OddsRatio",
  fun.aggregate = max, fill = 0)
# Replace Inf with the finite maximum
heatmap_targets[is.infinite(heatmap_targets)] = max(
  heatmap_targets[!is.infinite(heatmap_targets)])
heatmap_targets[heatmap_targets != 0] = log2(
  heatmap_targets[heatmap_targets != 0])
hc_targets = as.dist((1 - cor(heatmap_targets)) / 2)
hc_targets_line = as.dist((1 - cor(t(heatmap_targets))) / 2)

# hm_colorscale = colorRampPalette(
#   c('#b2182b', '#d6604d', '#f4a582', '#f7f7f7', '#f7f7f7', '#f7f7f7',
#     '#f7f7f7', '#f7f7f7', '#f7f7f7'))(150)

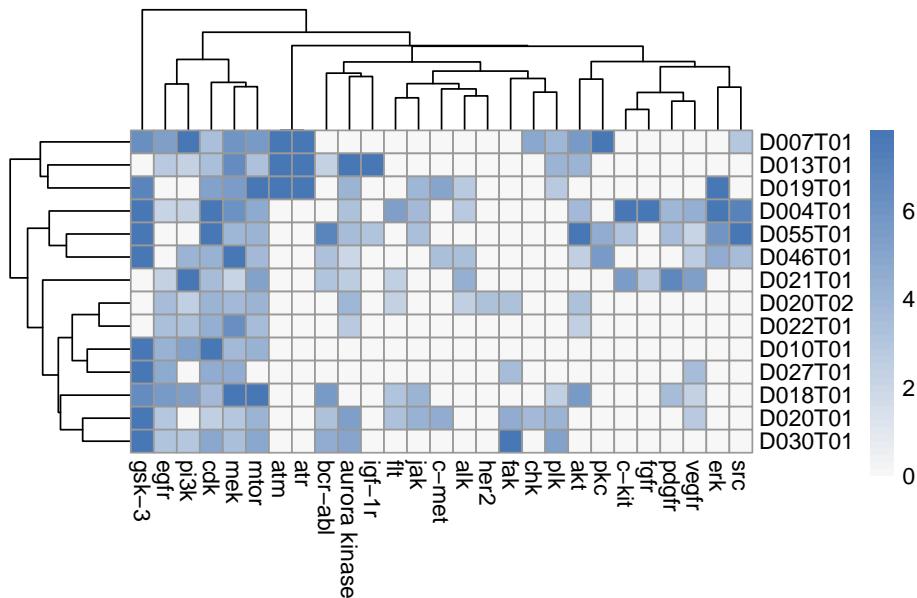
hm_breaks = seq(0, max(heatmap_pathways, na.rm = TRUE), length.out = 50)
hm_colorscale = colorRampPalette(c("#f7f7f7", "#4575B4"))(length(hm_breaks))
pheatmap(
  heatmap_pathways, cluster_rows = hc_pathways_line,
  cluster_cols = hc_pathways,
  breaks = hm_breaks, color = hm_colorscale,
  cellwidth = 10, cellheight = 10)
```

Drug-Induced Phenotypes



```
if(save_images) {  
  pheatmap(  
    heatmap_pathways, cluster_rows = hc_pathways_line,  
    cluster_cols = hc_pathways,  
    breaks = hm_breaks, color = hm_colorscale,  
    cellwidth = 10, cellheight = 10,  
    filename = file.path(img_out_dir, "Heatmap_LogOdds_Pathways.pdf"))  
}  
  
hm_breaks = seq(0, max(heatmap_targets, na.rm = TRUE), length.out = 50)  
hm_colorscale = colorRampPalette(c("#f7f7f7", "#4575B4"))(length(hm_breaks))  
pheatmap(  
  heatmap_targets, cluster_rows = hc_targets_line,  
  cluster_cols = hc_targets,  
  breaks = hm_breaks, color = hm_colorscale,  
  cellwidth = 10, cellheight = 10)
```

Drug-Induced Phenotypes

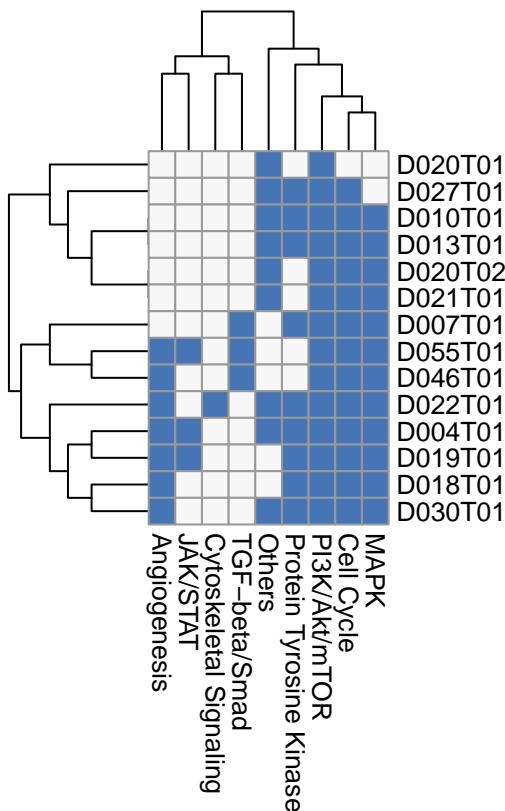


```
if(save_images) {  
  pheatmap(  
    heatmap_targets, cluster_rows = hc_targets_line,  
    cluster_cols = hc_targets,  
    breaks = hm_breaks, color = hm_colorscale,  
    cellwidth = 10, cellheight = 10,  
    filename = file.path(img_out_dir, "Heatmap_LogOdds_Targets.pdf"))  
}
```

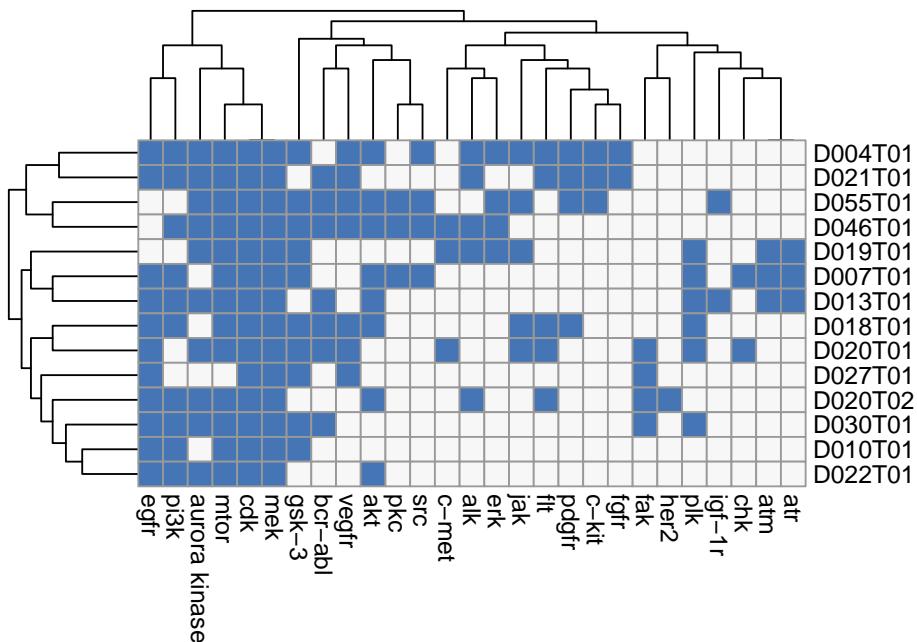
3.2.2 Binary Heatmap

```
binary_heatmap_pathways = heatmap_pathways  
binary_heatmap_pathways[binary_heatmap_pathways > 0] = 1  
binary_heatmap_targets = heatmap_targets  
binary_heatmap_targets[binary_heatmap_targets > 0] = 1  
  
hm_colorscale = colorRampPalette(c("#f7f7f7", "#4575B4))(2)  
pheatmap(  
  binary_heatmap_pathways, cluster_rows = hc_pathways_line,  
  cluster_cols = hc_pathways,  
  color = hm_colorscale, legend = FALSE,  
  cellwidth = 10, cellheight = 10)
```

Drug-Induced Phenotypes



```
pheatmap(
  binary_heatmap_targets, cluster_rows = hc_targets_line,
  cluster_cols = hc_targets,
  color = hm_colorscale, legend = FALSE,
  cellwidth = 10, cellheight = 10)
```



Drug-Induced Phenotypes

```
if(save_images) {
  pheatmap(
    binary_heatmap_pathways, cluster_rows = hc_pathways_line,
    cluster_cols = hc_pathways,
    color = hm_colorscale, legend = FALSE,
    cellwidth = 10, cellheight = 10,
    filename = file.path(img_out_dir, "Heatmap_Binary_Pathways.pdf"))
  pheatmap(
    binary_heatmap_targets, cluster_rows = hc_targets_line,
    cluster_cols = hc_targets,
    color = hm_colorscale, legend = FALSE,
    cellwidth = 10, cellheight = 10,
    filename = file.path(img_out_dir, "Heatmap_Binary_Targets.pdf"))
}
```

3.3 Feature Differences

I look at the vectors from median DMSO to median drug features for each drug per replicate. I use the well-averaged features to improve loading times. For each line, I aggregate all drugs belonging to the enriched clusters.

```
# Extract only key features
key_features = c(
  "x.0.s.area_expected" = "Area",
  "x.0.m.eccentricity_expected" = "Eccentricity",
  "x.a.b.q05_expected" = "Median Actin Intensity",
  "x.b.b.q05_expected" = "Median FITC Intensity",
  "x.c.b.q05_expected" = "Median DAPI Intensity")

# Set up color scale
pheno_color_scale = c(
  "Median Actin Intensity" = "#e6194b",
  "Median FITC Intensity" = "#3cb44b",
  "Median DAPI Intensity" = "#0082c8",
  "Area" = "#ffe119",
  "Eccentricity" = "#f58231")

# Load features
data("well_features", package = "SCOPEAnalysis")
well_features = well_features[, names(key_features)]

# Z-Scale features for each line and replicate
for(line in unique(well_metadata$Line)) {
  for(r in unique(well_metadata$Replicate)) {
    rep_sel = well_metadata$Line == line & well_metadata$Replicate == r
    well_features[rep_sel, ] = apply(
      X = well_features[rep_sel, ],
      MARGIN = 2, FUN = function(x) (x - median(x)) / mad(x))
  }
}
```

Drug-Induced Phenotypes

```
}

# Subtract DMSO medians for each line and replicate
for(line in unique(well_metadata$Line)) {
  for(r in unique(well_metadata$Replicate)) {
    rep_sel = well_metadata$Line == line & well_metadata$Replicate == r
    rep_line_dmso_median = apply(
      X = well_features[rep_sel & well_metadata$Drug == "DMSO", ],
      MARGIN = 2, FUN = median, na.rm = TRUE)
    well_features[rep_sel, ] = sweep(
      x = well_features[rep_sel, ],
      MARGIN = 2, STATS = rep_line_dmso_median, FUN = "-")
  }
}

# Aggregate features for pathways for each line and replicate
all_pathway_features = list()
all_target_features = list()
for(line in unique(well_metadata$Line)) {
  line_enriched_drugs = enriched_drugs[enriched_drugs$Line == line, ]
  line_enriched_pathways = unique(line_enriched_drugs[
    line_enriched_drugs$is.pathway.enriched, "Pathway"])
  line_enriched_targets = unique(line_enriched_drugs[
    line_enriched_drugs$is.target.enriched, "Enriched.Target"])

  # Pathways
  for(pathway in line_enriched_pathways) {
    drugs = line_enriched_drugs[
      line_enriched_drugs$Pathway == pathway &
      line_enriched_drugs$is.pathway.enriched, ]
    sel_indices = well_metadata$Drug %in% drugs$Drug & well_metadata$Line == line
    relevant_features = well_features[sel_indices, ]
    relevant_metadata = well_metadata[sel_indices, ]
    # Aggregate
    pathway_features = aggregate(
      x = relevant_features,
      by = list("Replicate" = relevant_metadata$Replicate),
      FUN = median)
    pathway_features$Line = line
    pathway_features$Label = pathway
    all_pathway_features[[length(all_pathway_features) + 1]] = pathway_features
  }

  # Targets
  for(target in line_enriched_targets) {
    drugs = line_enriched_drugs[
      line_enriched_drugs$Enriched.Target == target &
      line_enriched_drugs$is.target.enriched, ]
    sel_indices = well_metadata$Drug %in% drugs$Drug & well_metadata$Line == line
    relevant_features = well_features[sel_indices, ]
    relevant_metadata = well_metadata[sel_indices, ]
```

Drug-Induced Phenotypes

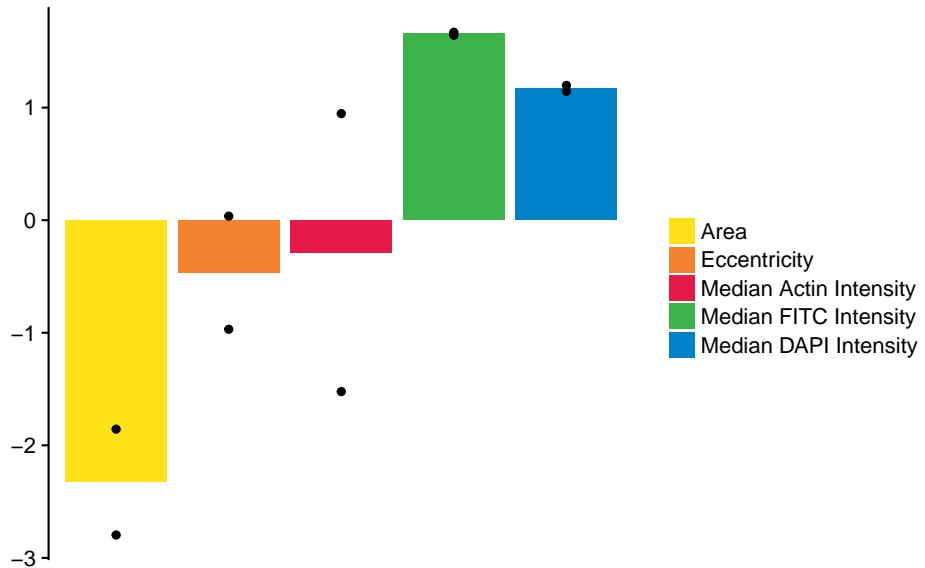
```
# Aggregate
target_features = aggregate(
  x = relevant_features,
  by = list("Replicate" = relevant_metadata$Replicate),
  FUN = median)
target_features$Line = line
target_features$Label = target
all_target_features[[length(all_target_features) + 1]] = target_features
}
}
all_pathway_features = do.call(rbind, all_pathway_features)
all_target_features = do.call(rbind, all_target_features)
```

3.3.1 Phenoprints for Pathways

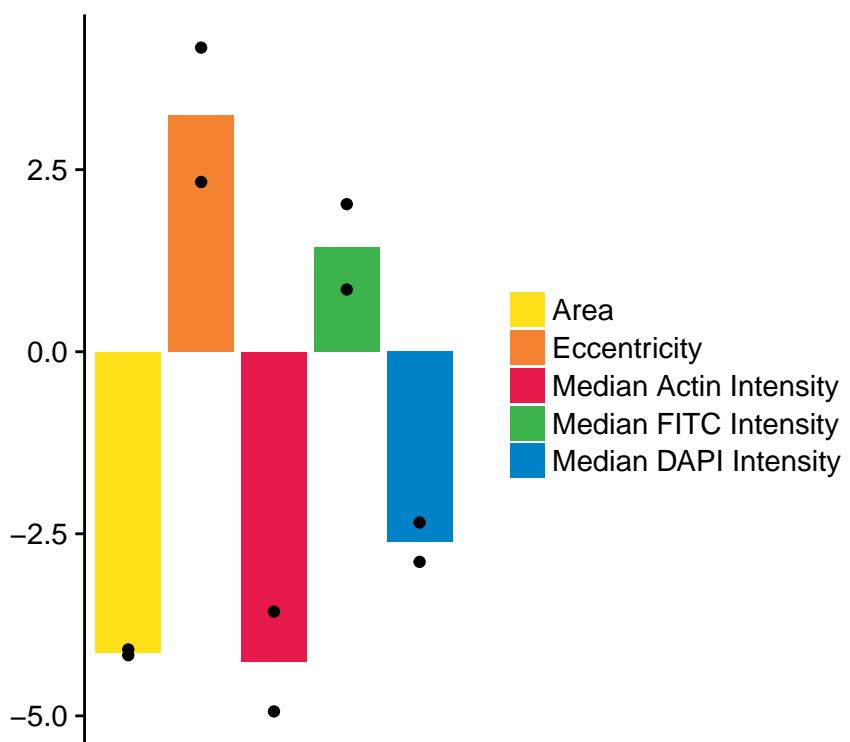
```
for(line in unique(all_pathway_features$Line)) {
  labels = unique(all_pathway_features[
    all_pathway_features$Line == line, "Label"])
  for(label in labels) {
    ggplot_df = all_pathway_features[all_pathway_features$Line == line &
      all_pathway_features$Label == label, ]
    ggplot_df$Line = NULL
    ggplot_df$Label = NULL
    ggplot_df = melt(ggplot_df, id.vars = c("Replicate"))
    ggplot_df$variable = key_features[ggplot_df$variable]
    ggplot_df$variable = factor(ggplot_df$variable, levels = key_features)
    ggplot_df_bar = aggregate(
      x = ggplot_df$value,
      by = list("variable" = ggplot_df$variable),
      FUN = median)
    colnames(ggplot_df_bar) = c("variable", "value")
    gp = ggplot(mapping = aes(x = variable, y = value)) +
      geom_col(data = ggplot_df_bar, mapping = aes(fill = variable)) +
      geom_point(data = ggplot_df) +
      scale_fill_manual(values = pheno_color_scale) +
      theme_vignette() + coord_fixed() +
      theme(
        axis.text.x = element_blank(), axis.line.x = element_blank(),
        axis.ticks.x = element_blank(), legend.title = element_blank()) +
      xlab("") + ylab("") + ggtitle(paste0(line, " ", label))
    plot(gp)
    if(save_images) ggsave(
      filename = file.path(img_out_dir, sprintf(
        "Phenoprints_Pathways_%s_%s.pdf", line, make.names(label))),
      width = 5, height = 5, useDingbats = FALSE)
  }
}
```

Drug-Induced Phenotypes

D004T01 PI3K/Akt/mTOR

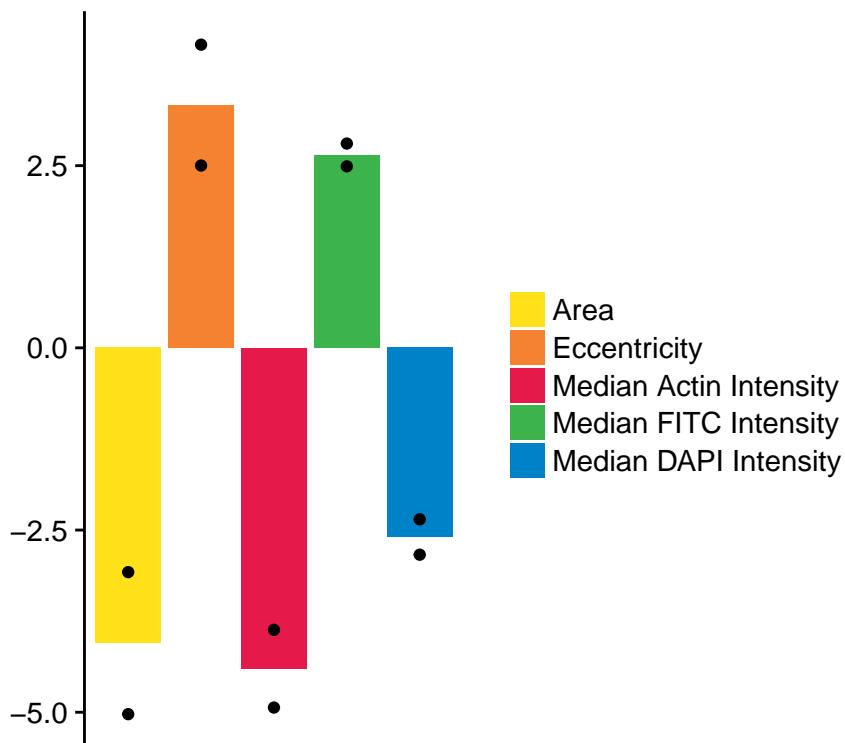


D004T01 Cell Cycle

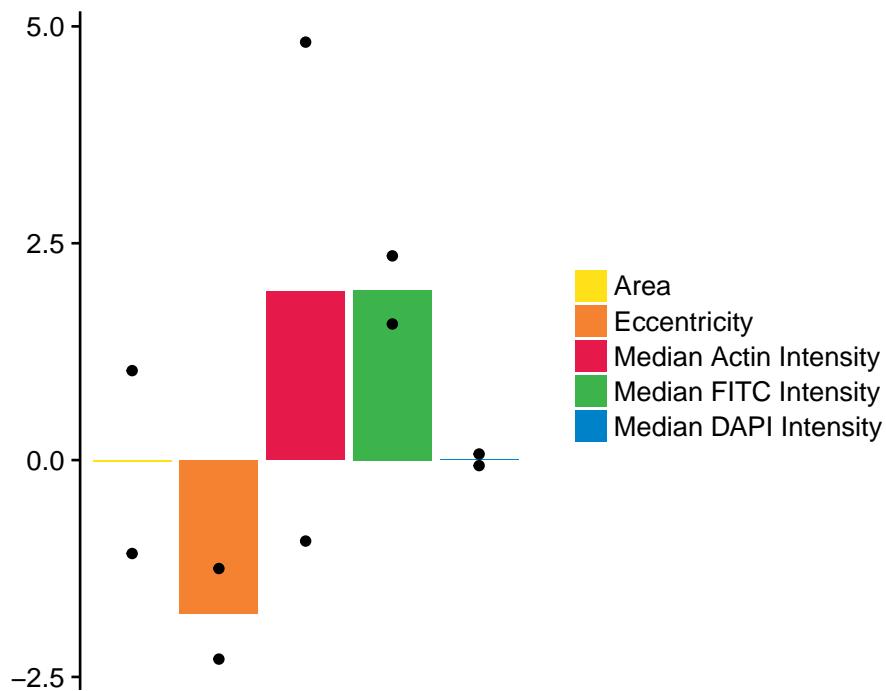


Drug-Induced Phenotypes

D004T01 MAPK

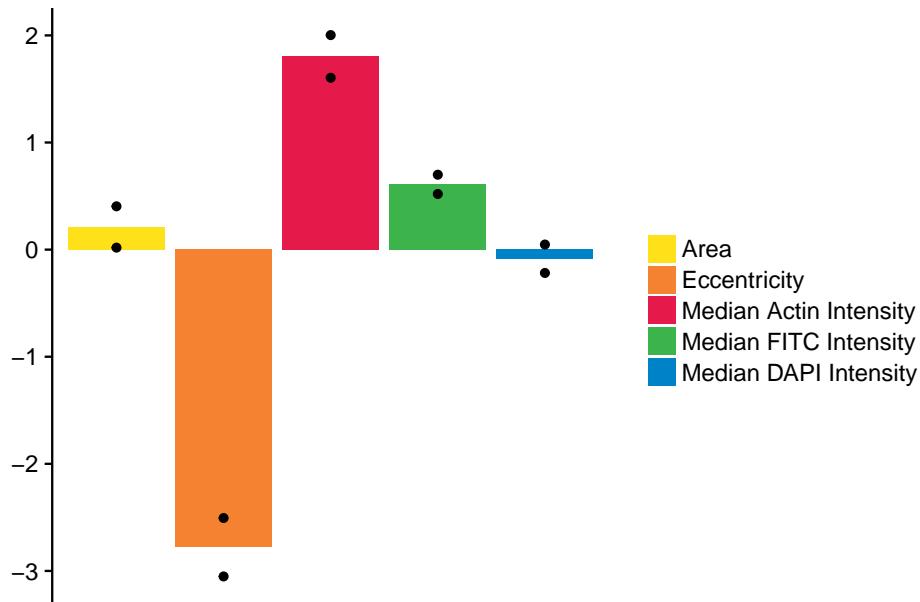


D004T01 Protein Tyrosine Kinase

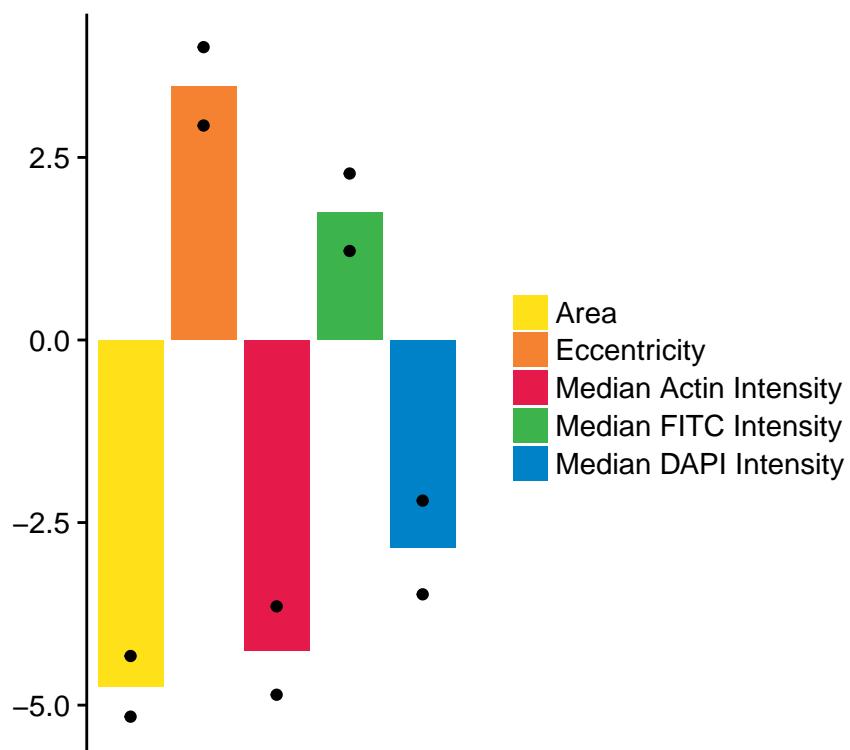


Drug-Induced Phenotypes

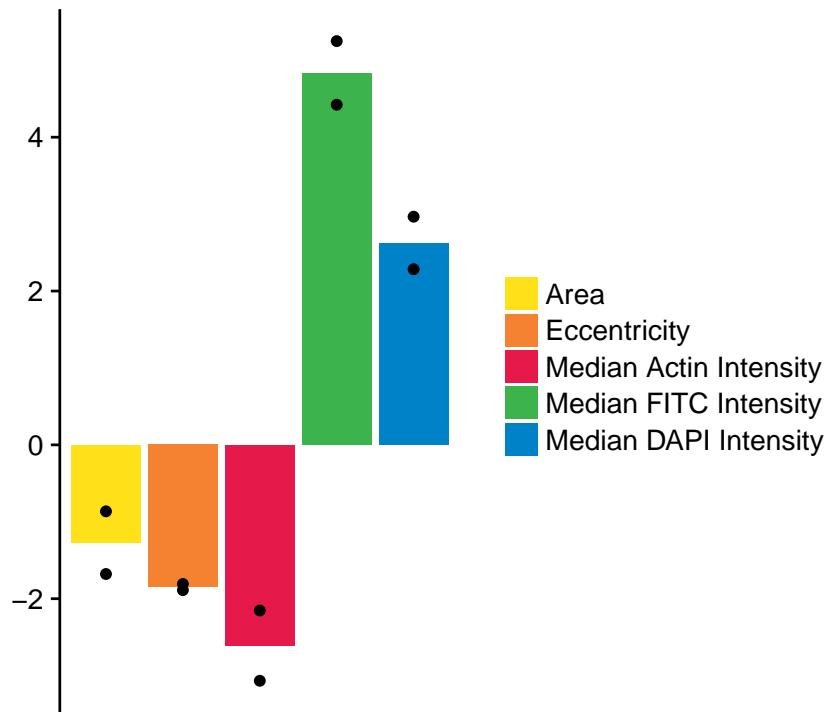
D004T01 Angiogenesis



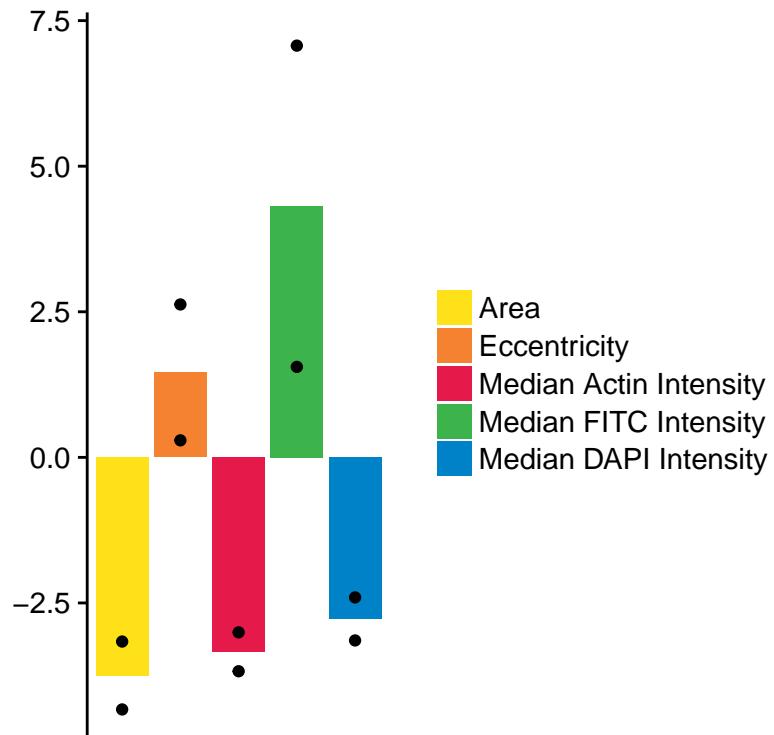
D004T01 JAK/STAT



D007T01 Protein Tyrosine Kinase

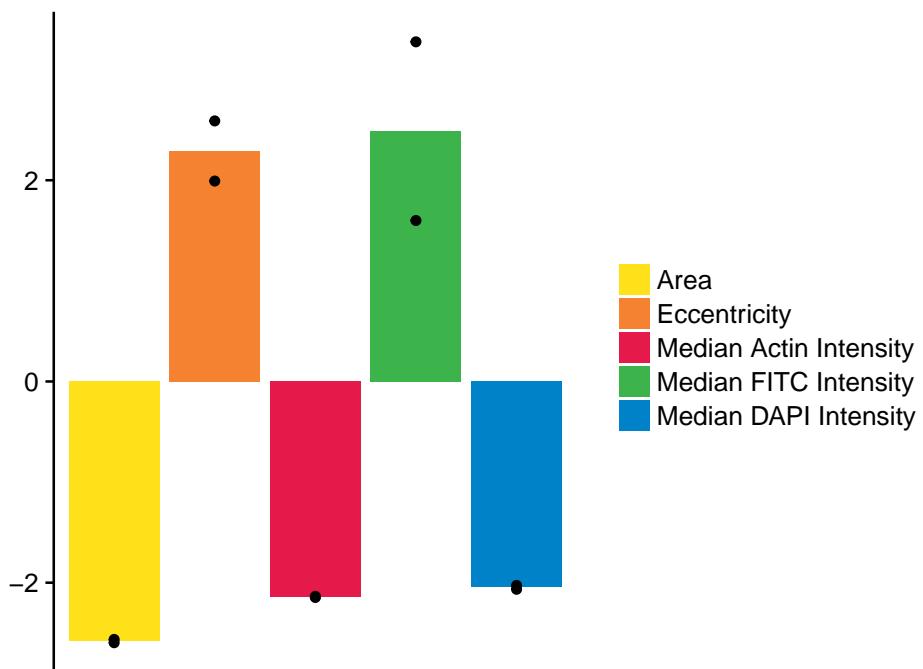


D007T01 Cell Cycle

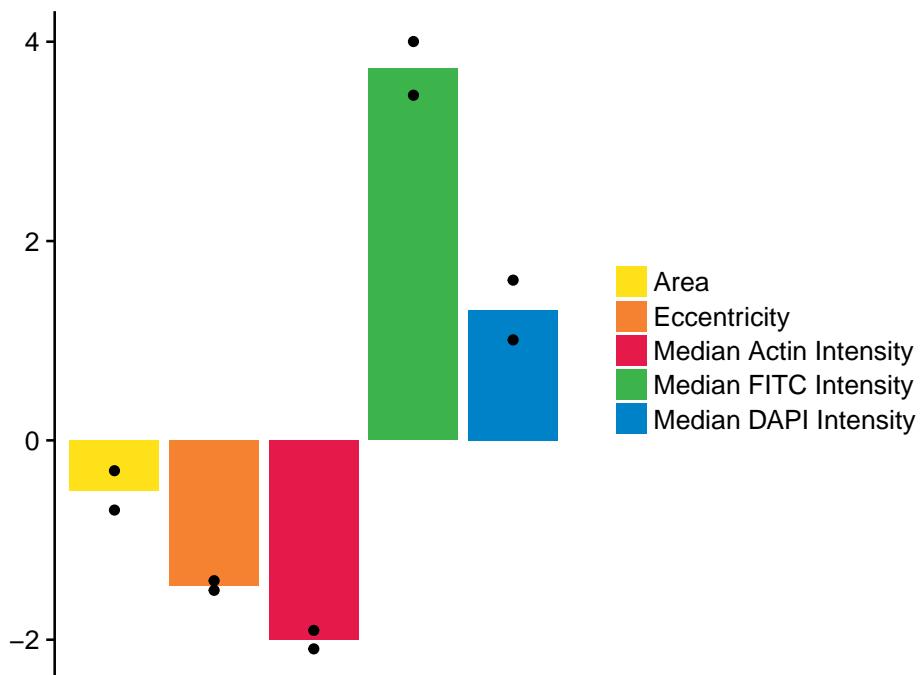


Drug-Induced Phenotypes

D007T01 PI3K/Akt/mTOR

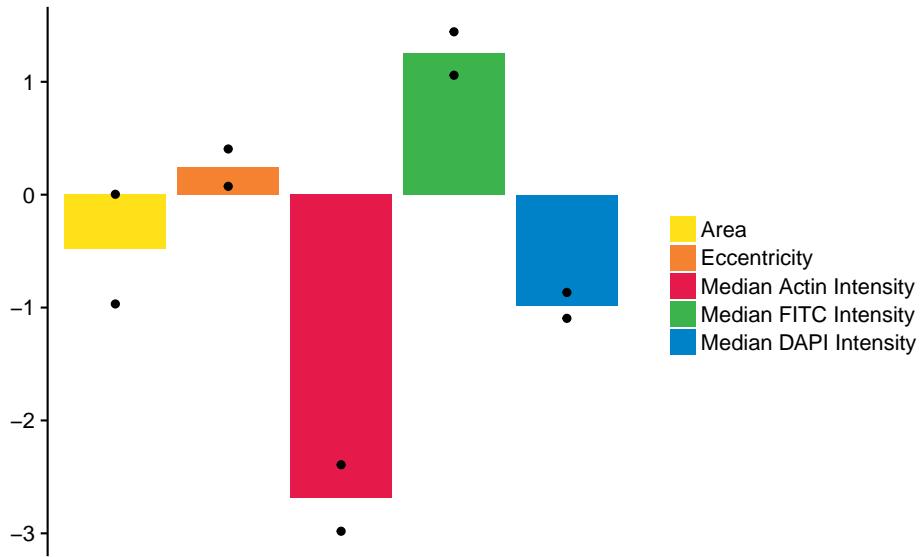


D007T01 MAPK

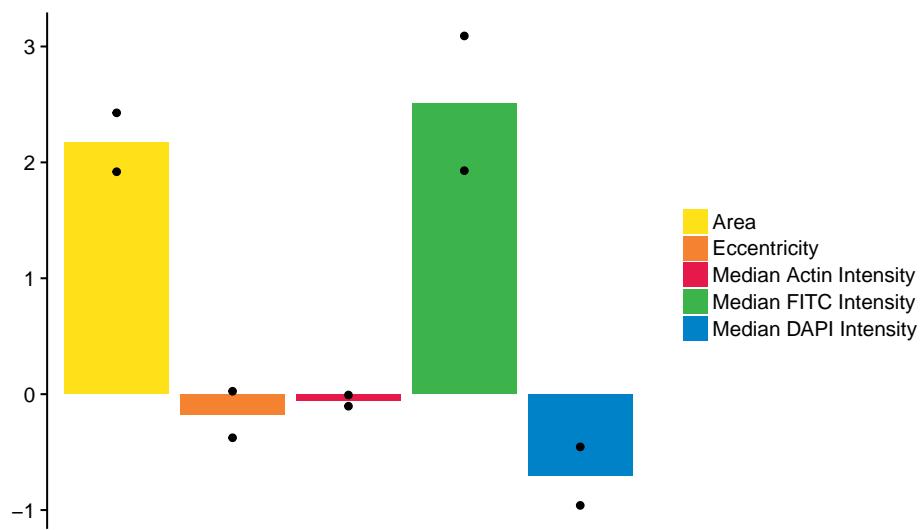


Drug-Induced Phenotypes

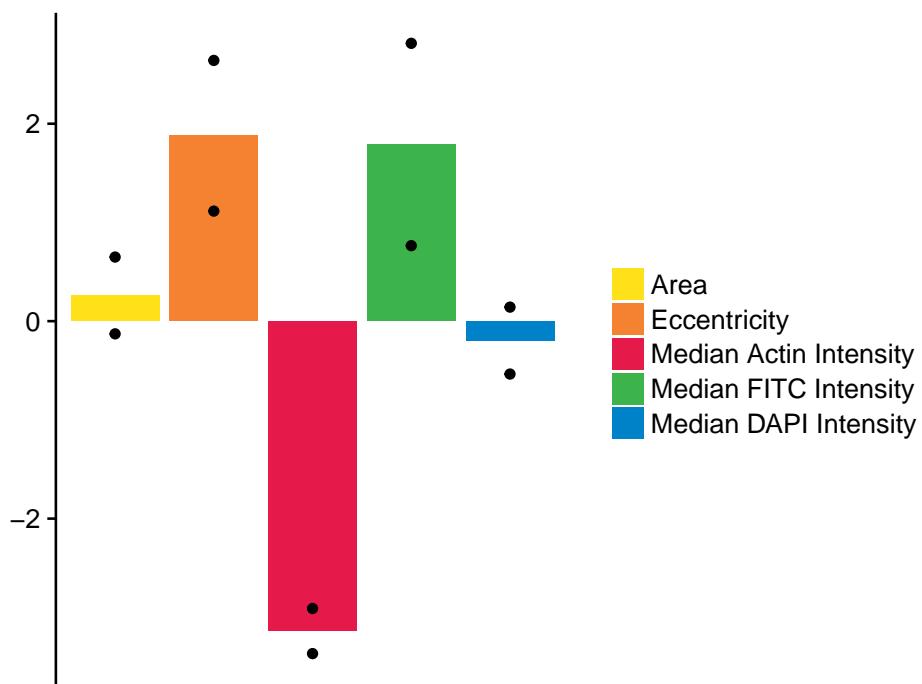
D007T01 TGF-beta/Smad



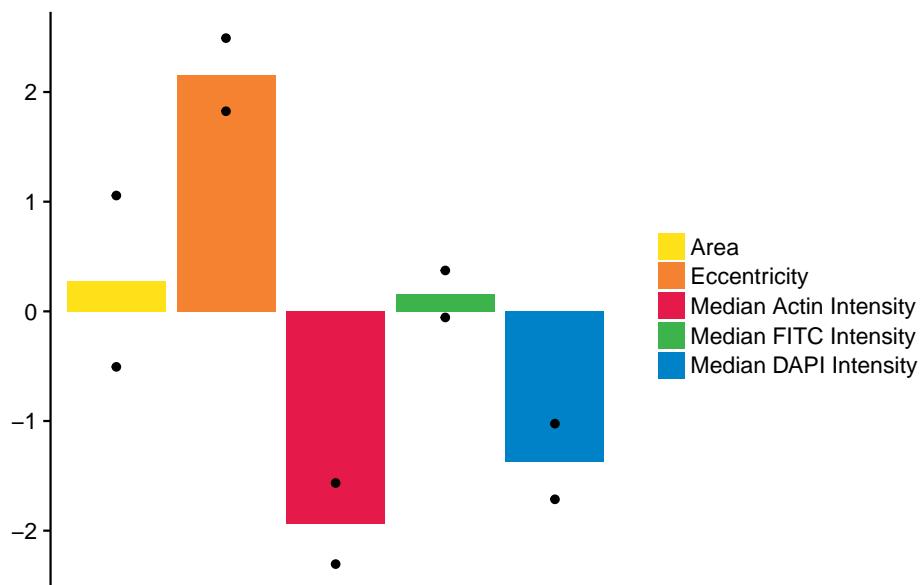
D010T01 Protein Tyrosine Kinase



D010T01 Cell Cycle

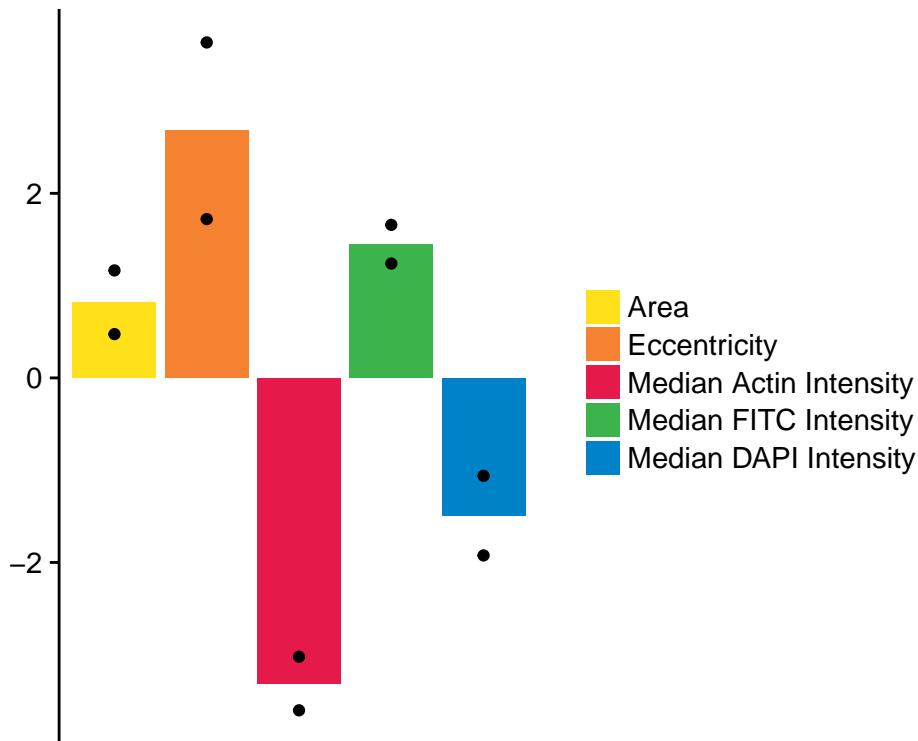


D010T01 PI3K/Akt/mTOR

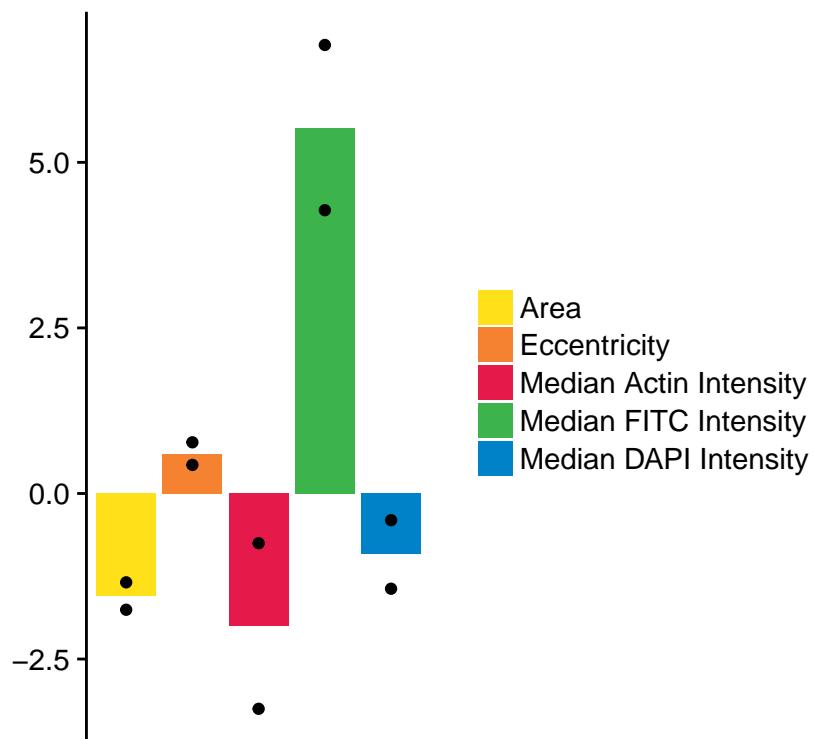


Drug-Induced Phenotypes

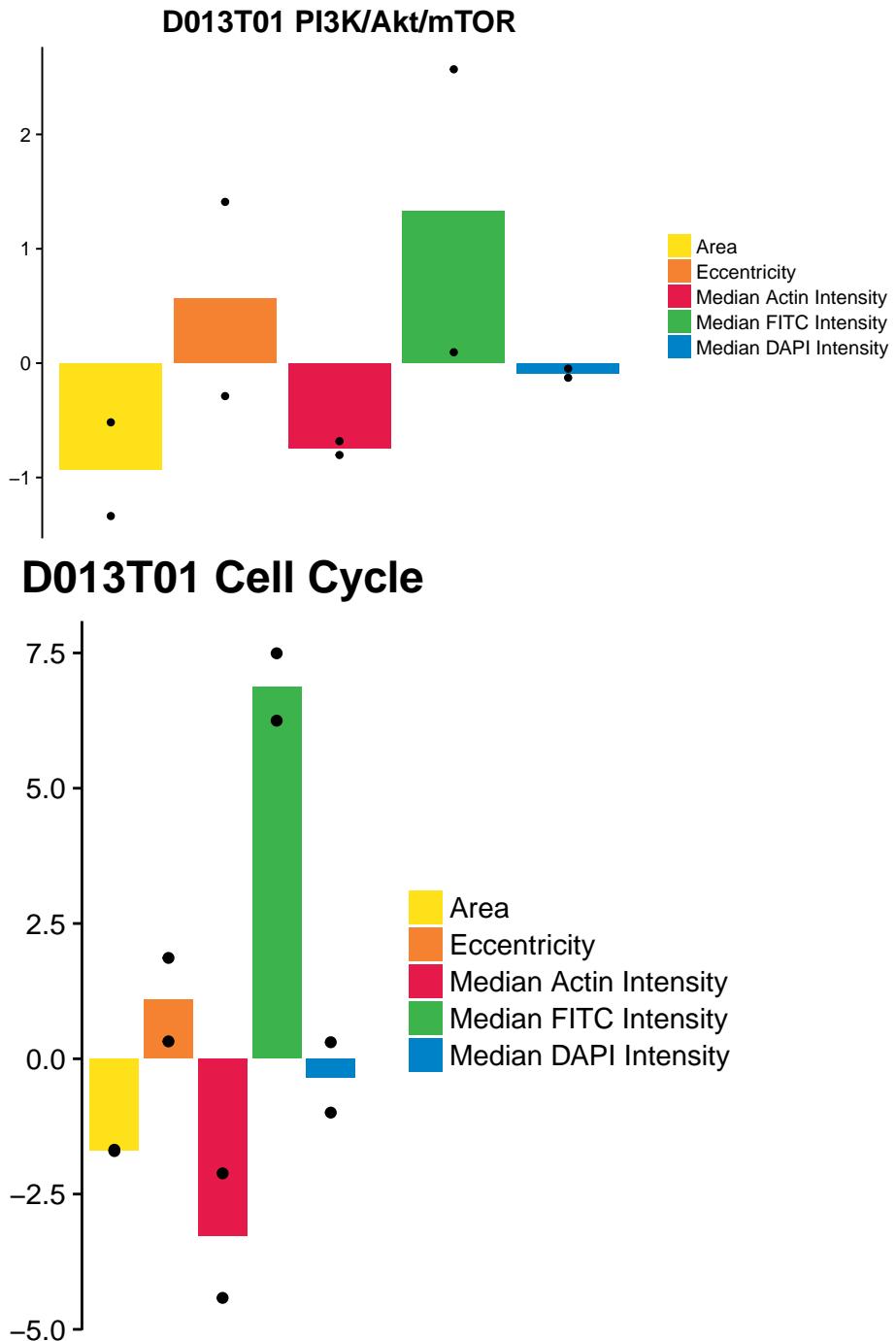
D010T01 MAPK



D013T01 Protein Tyrosine Kinase

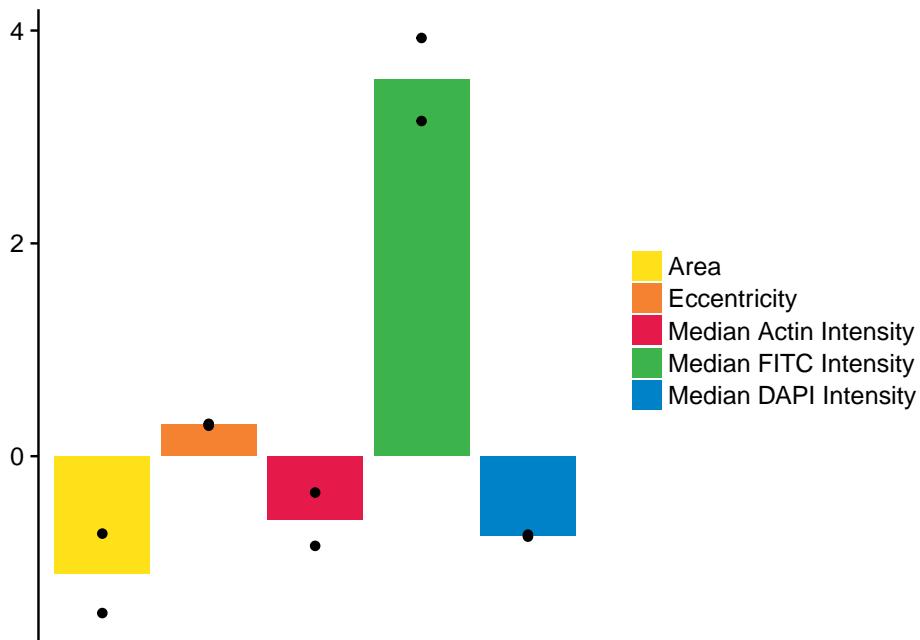


Drug-Induced Phenotypes

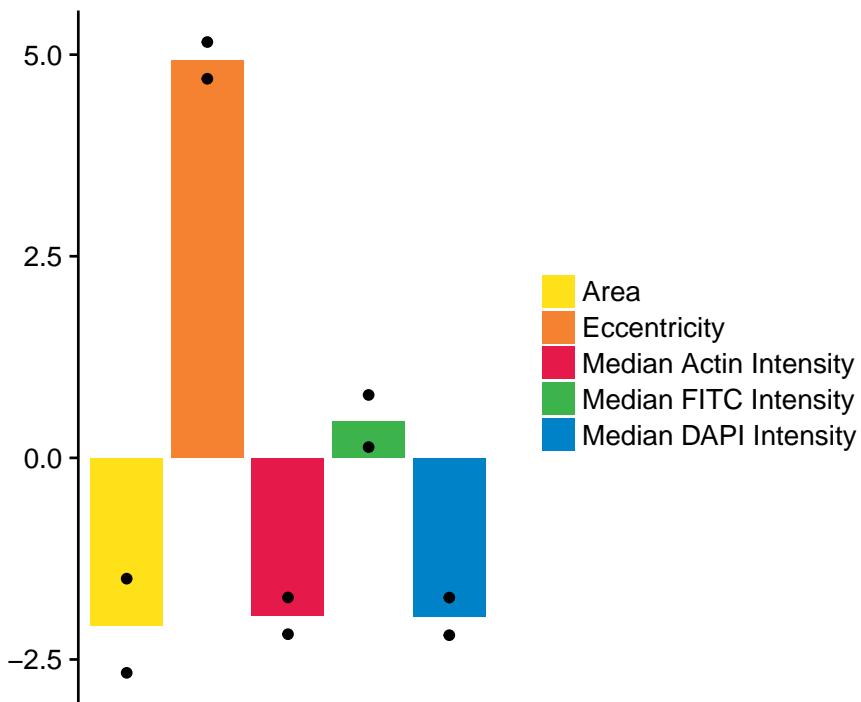


Drug-Induced Phenotypes

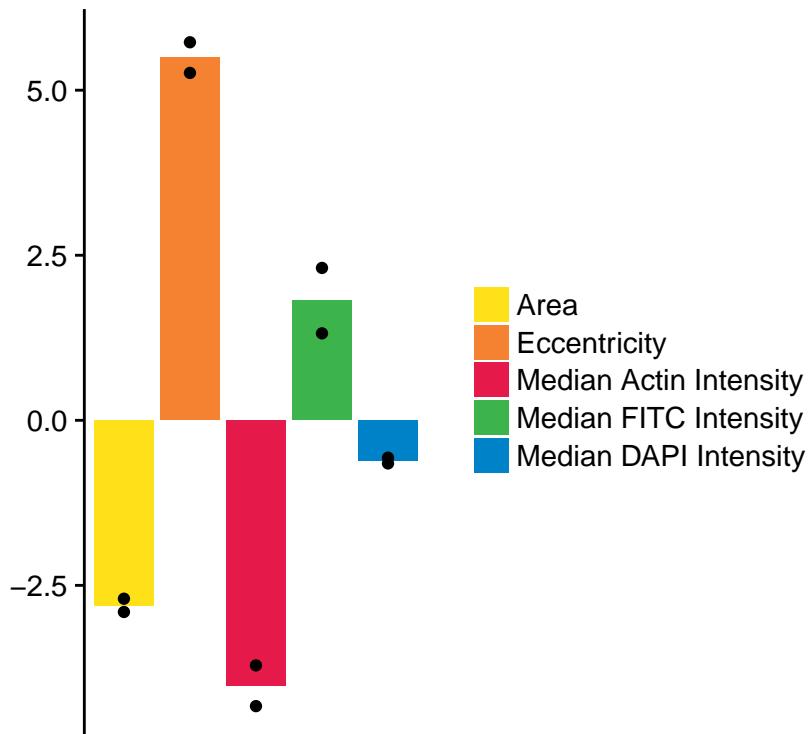
D013T01 MAPK



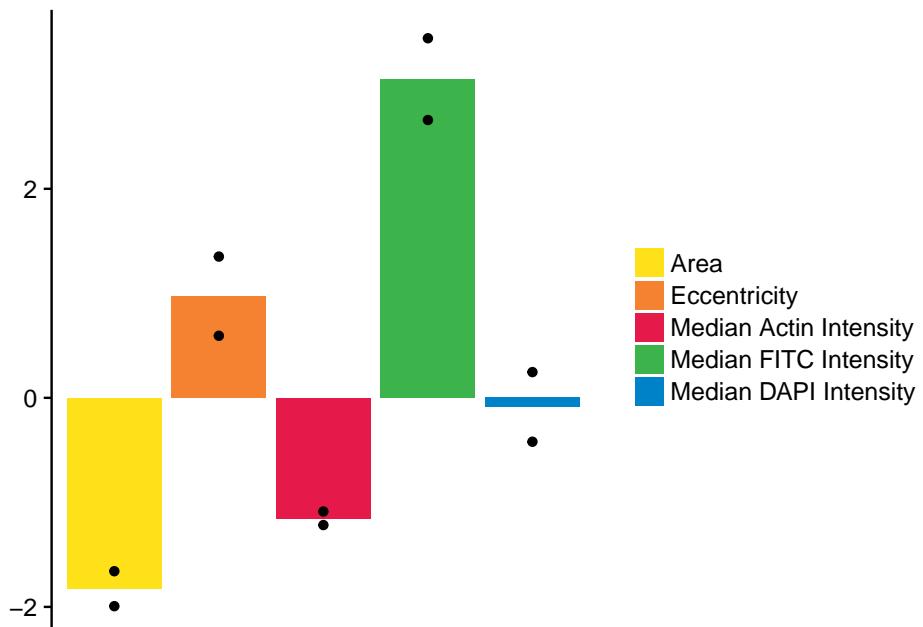
D018T01 Protein Tyrosine Kinase



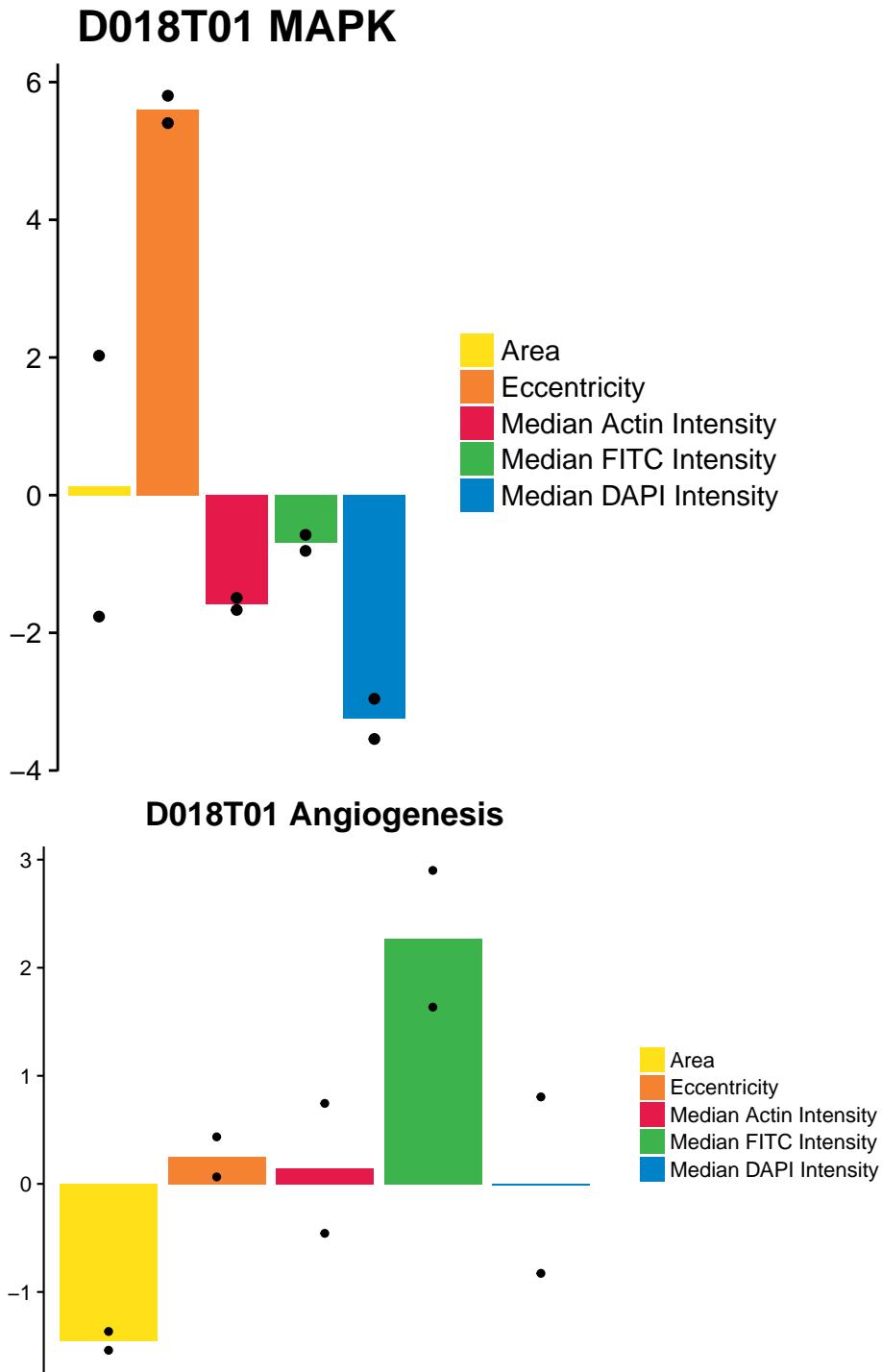
D018T01 Cell Cycle



D018T01 PI3K/Akt/mTOR

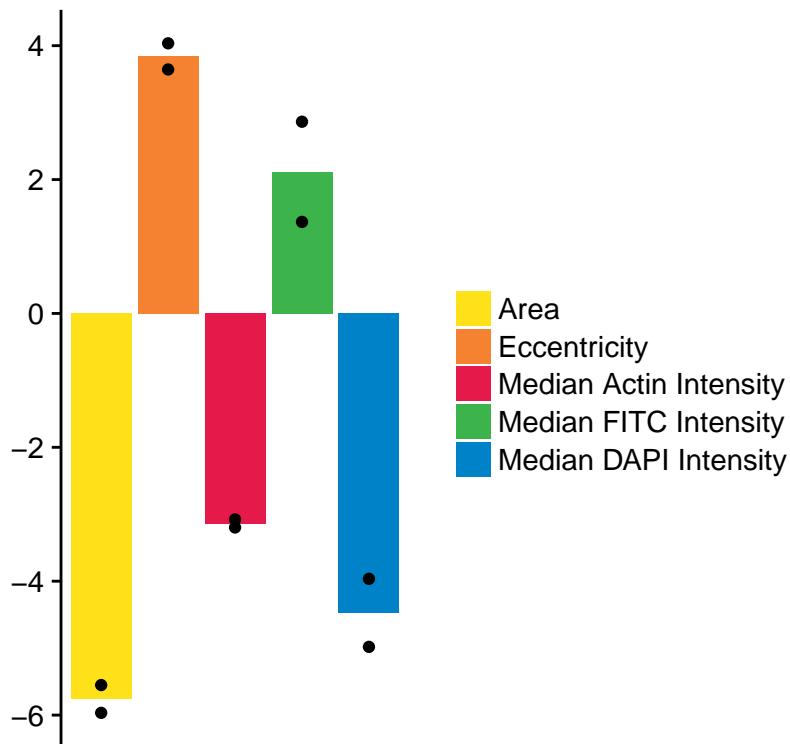


Drug-Induced Phenotypes

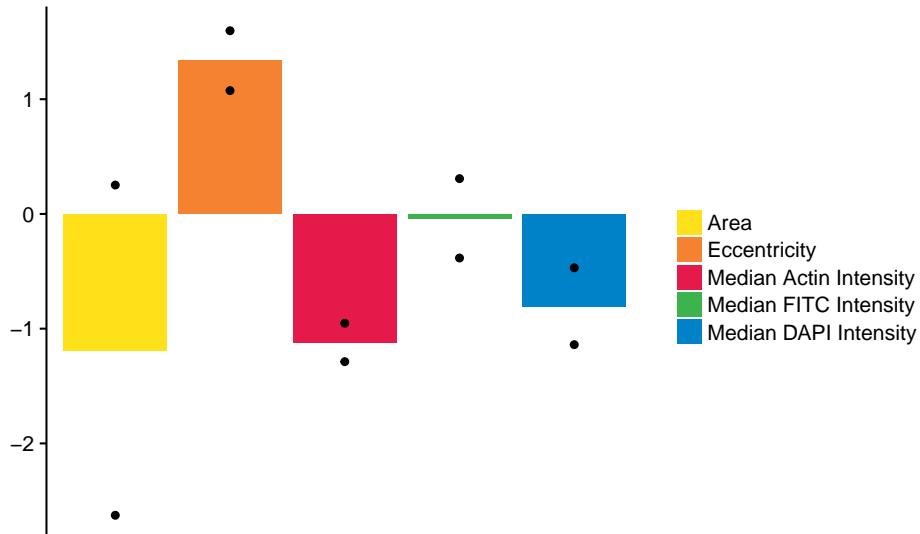


Drug-Induced Phenotypes

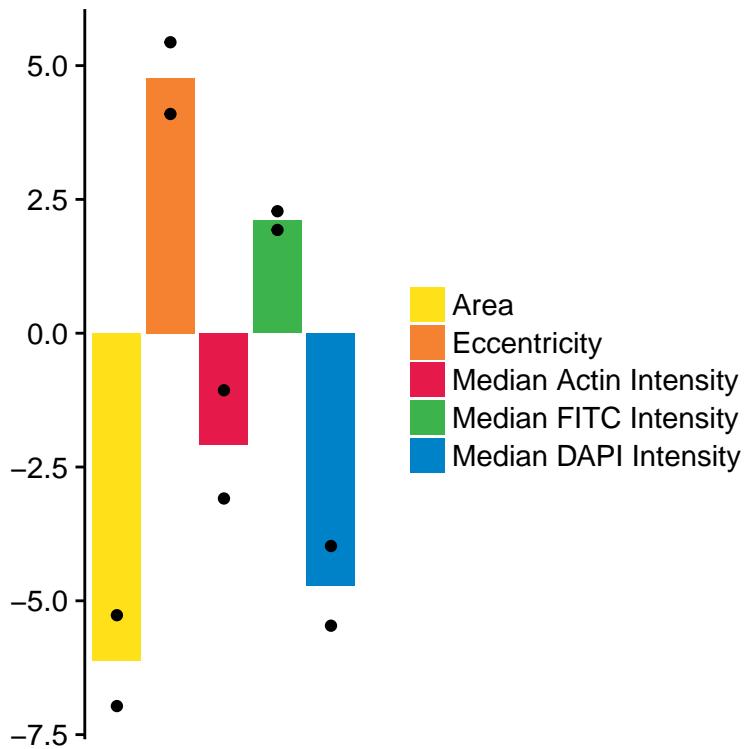
D019T01 JAK/STAT



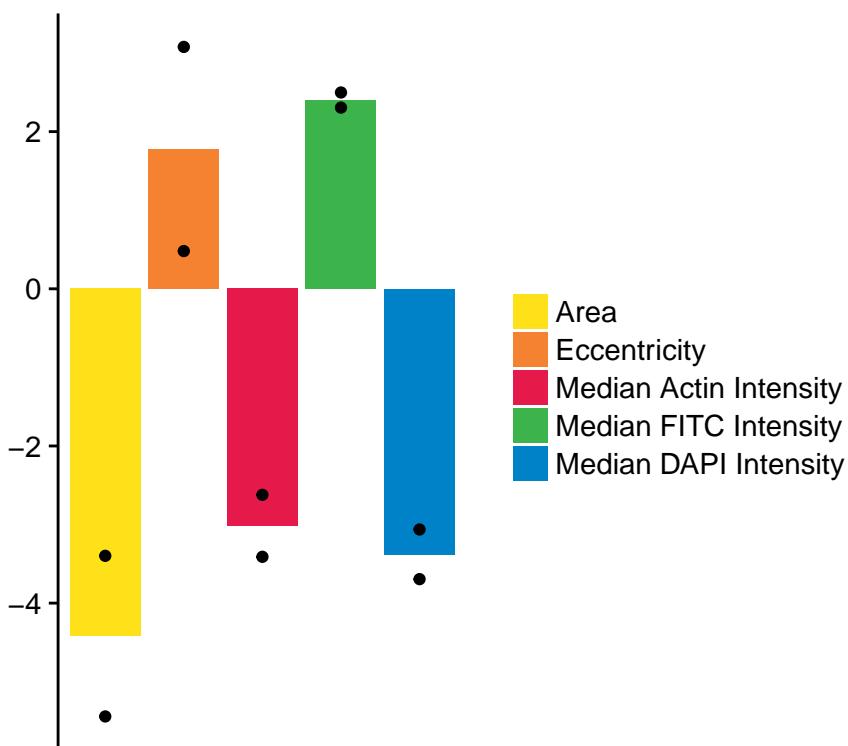
D019T01 PI3K/Akt/mTOR



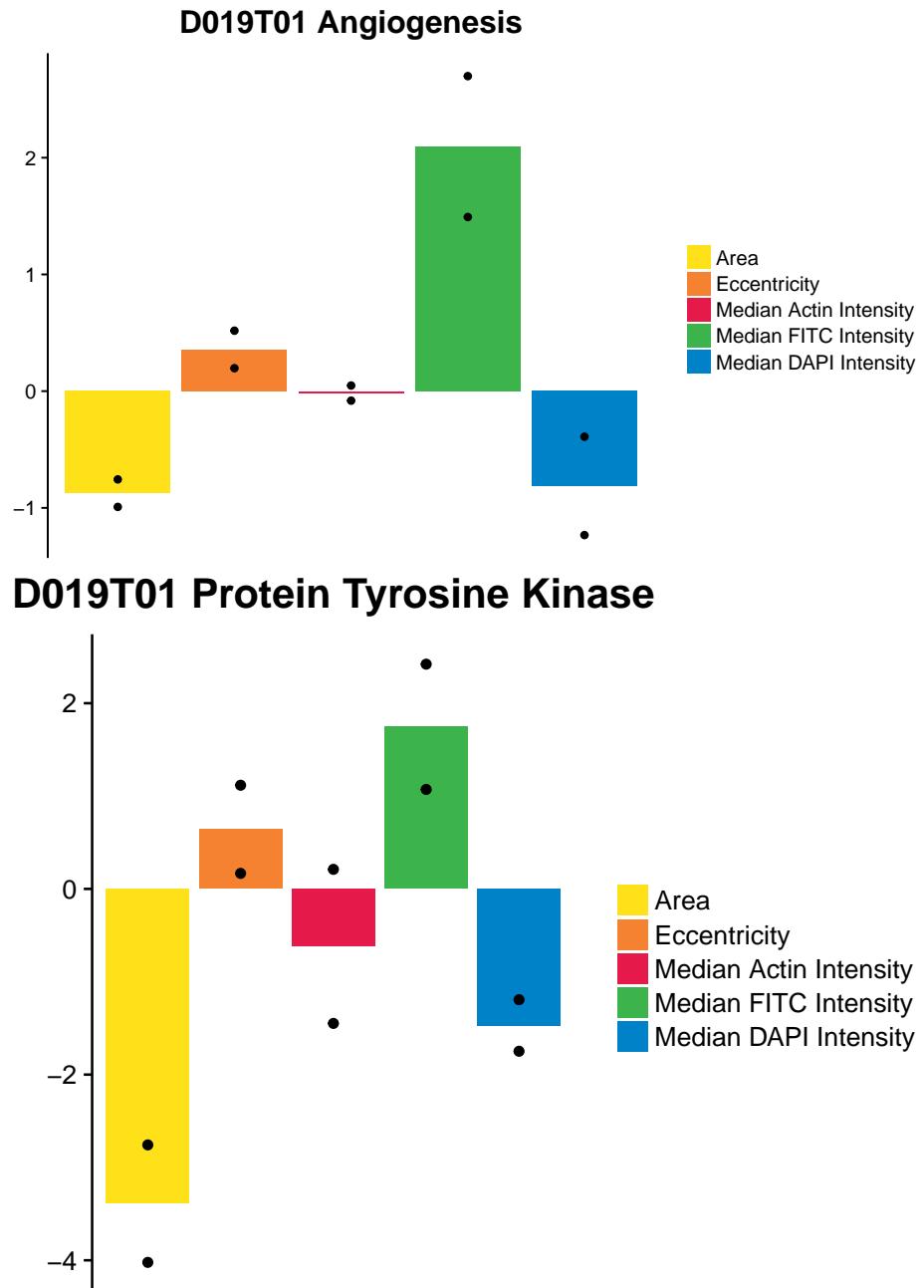
D019T01 Cell Cycle



D019T01 MAPK

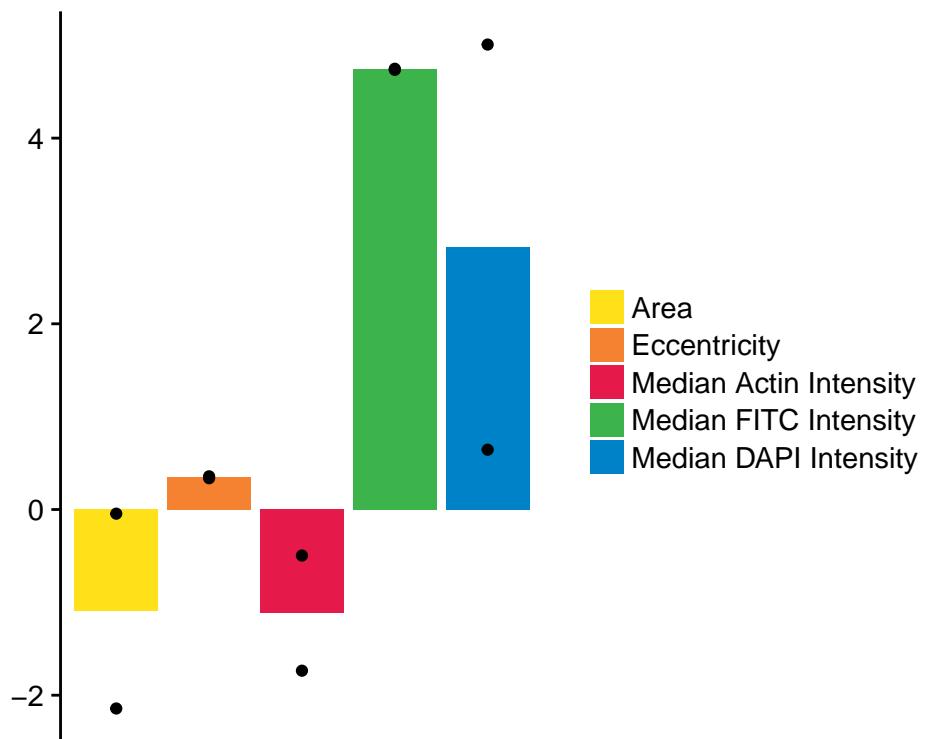


Drug-Induced Phenotypes

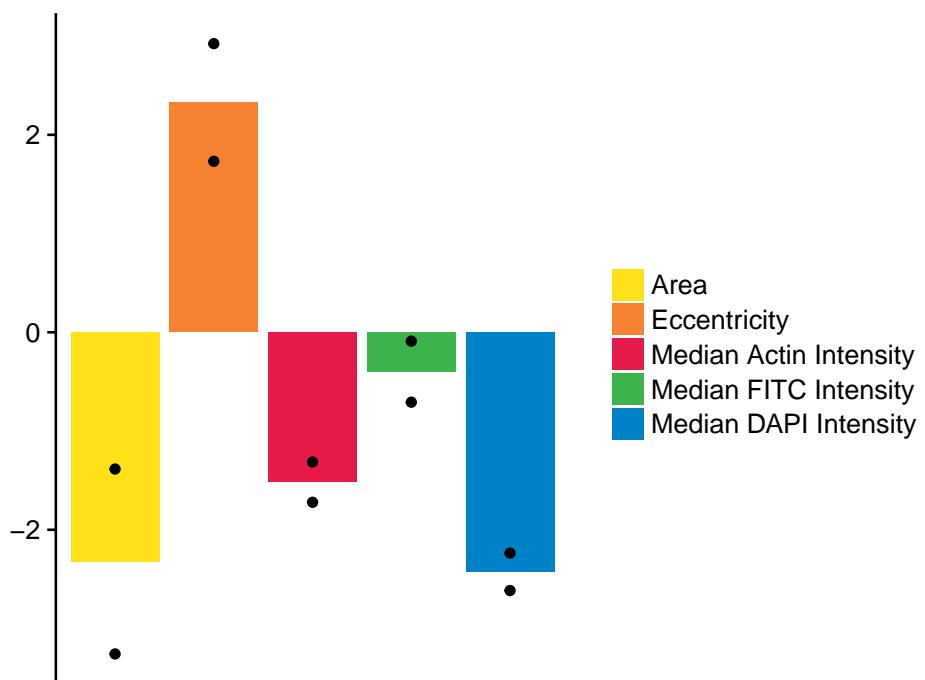


Drug-Induced Phenotypes

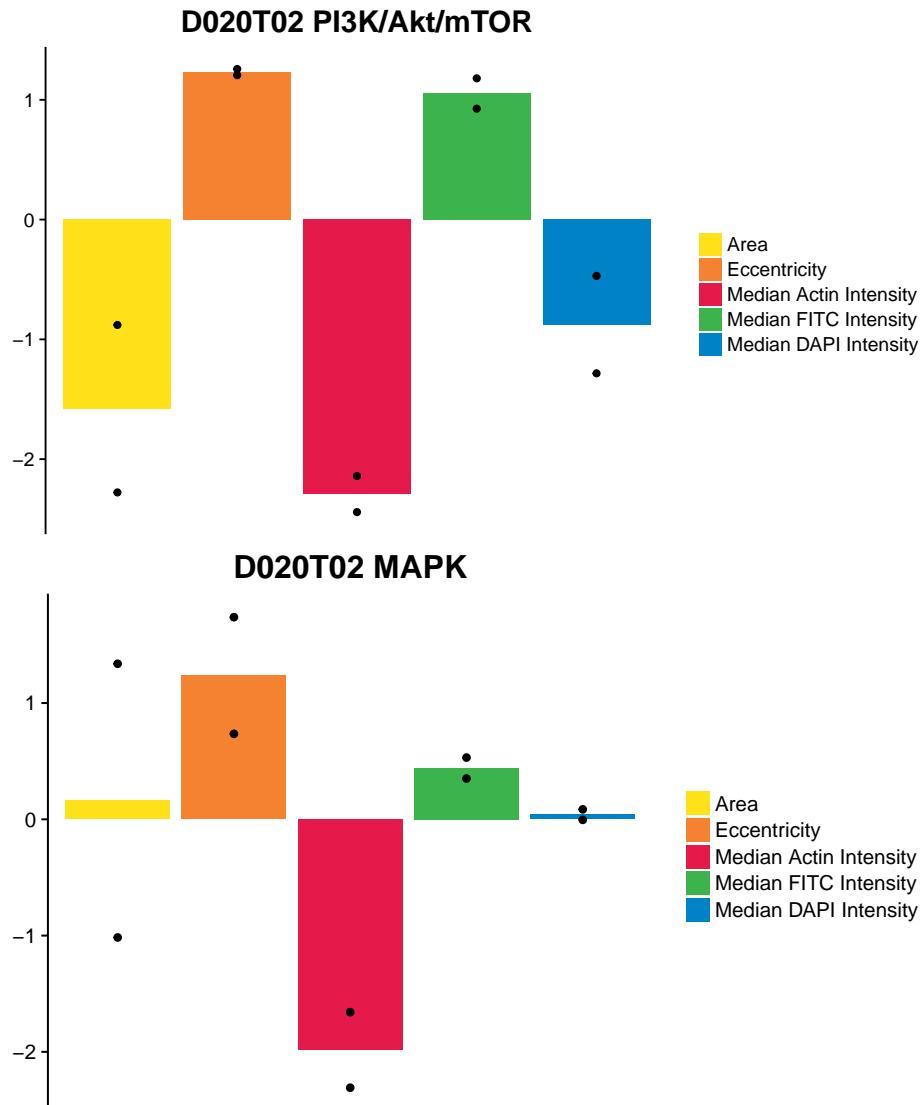
D020T01 PI3K/Akt/mTOR



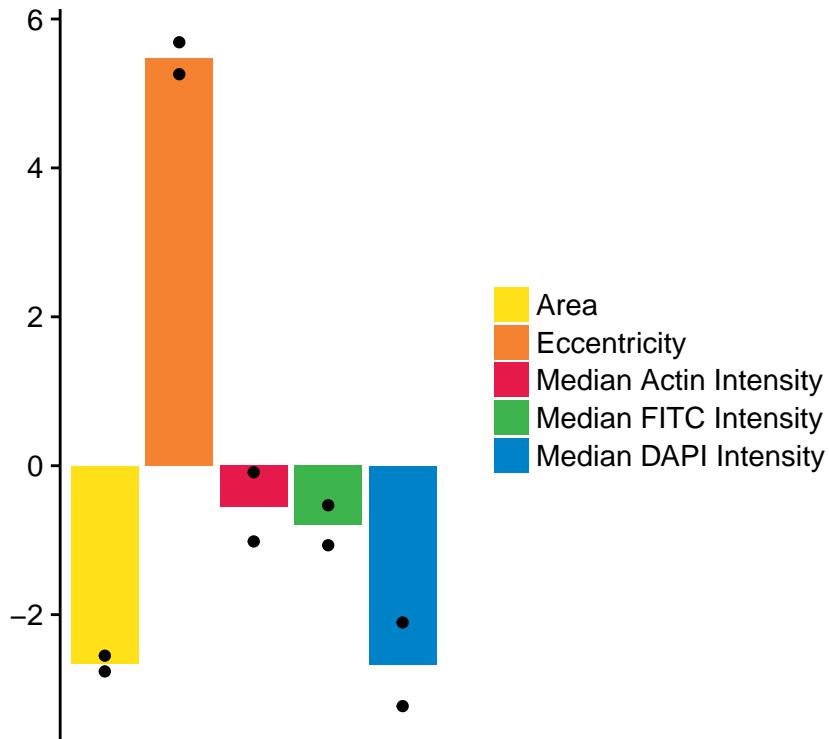
D020T02 Cell Cycle



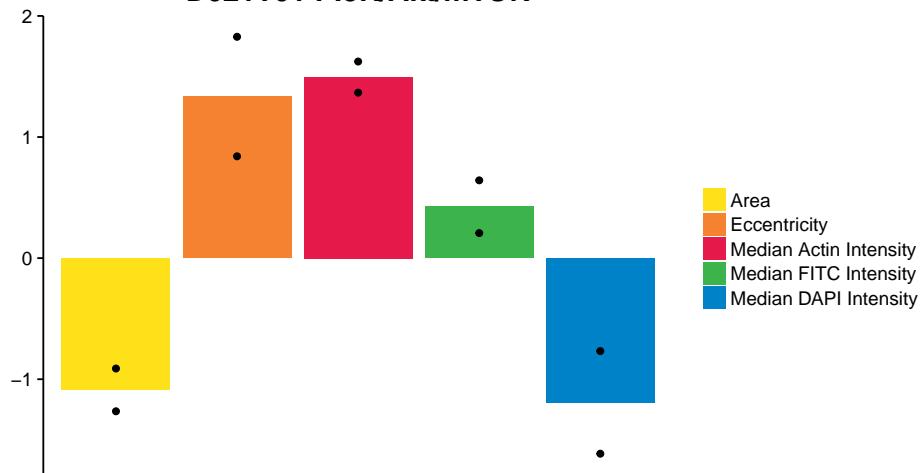
Drug-Induced Phenotypes



D021T01 Cell Cycle

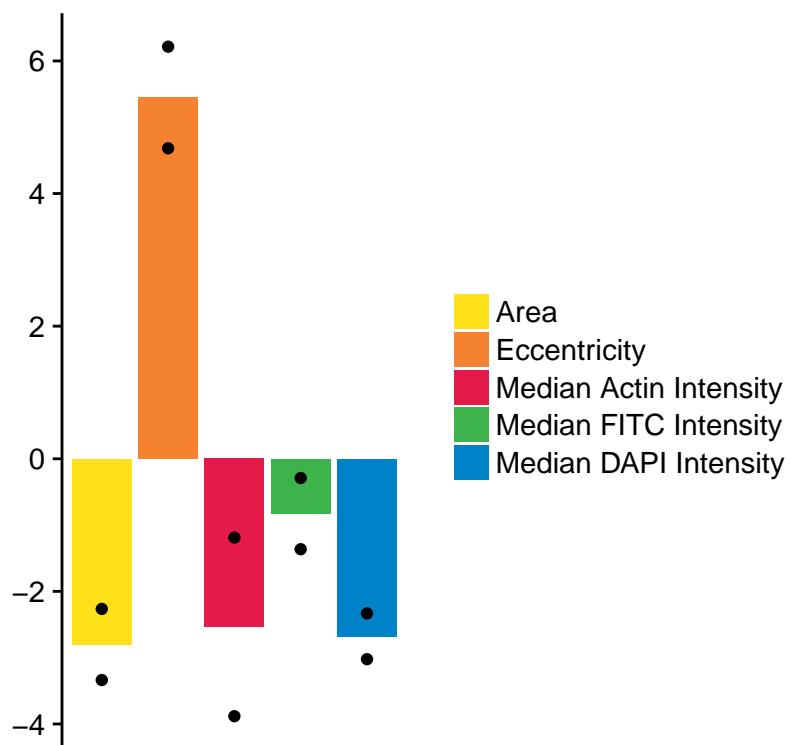


D021T01 PI3K/Akt/mTOR

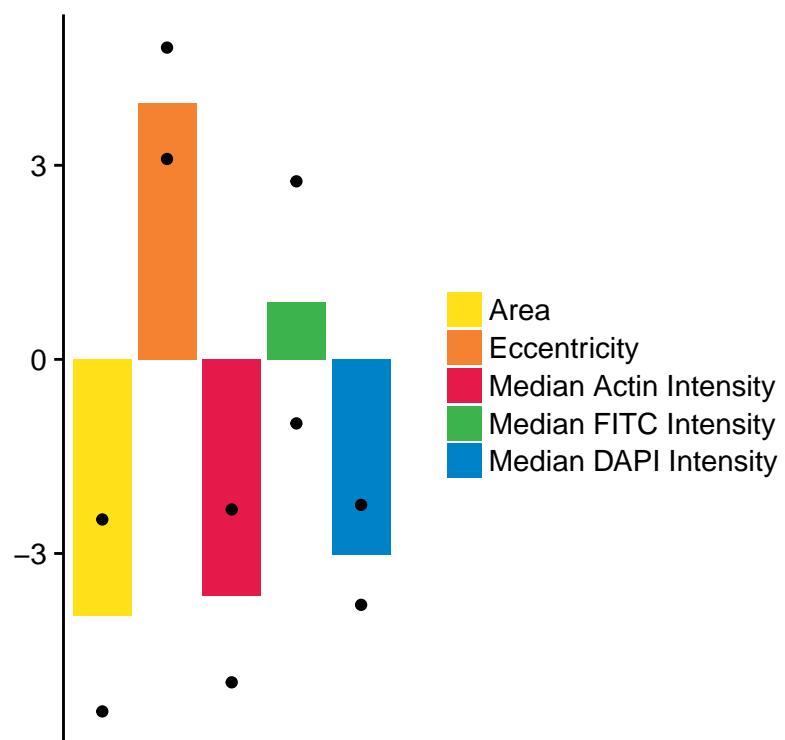


Drug-Induced Phenotypes

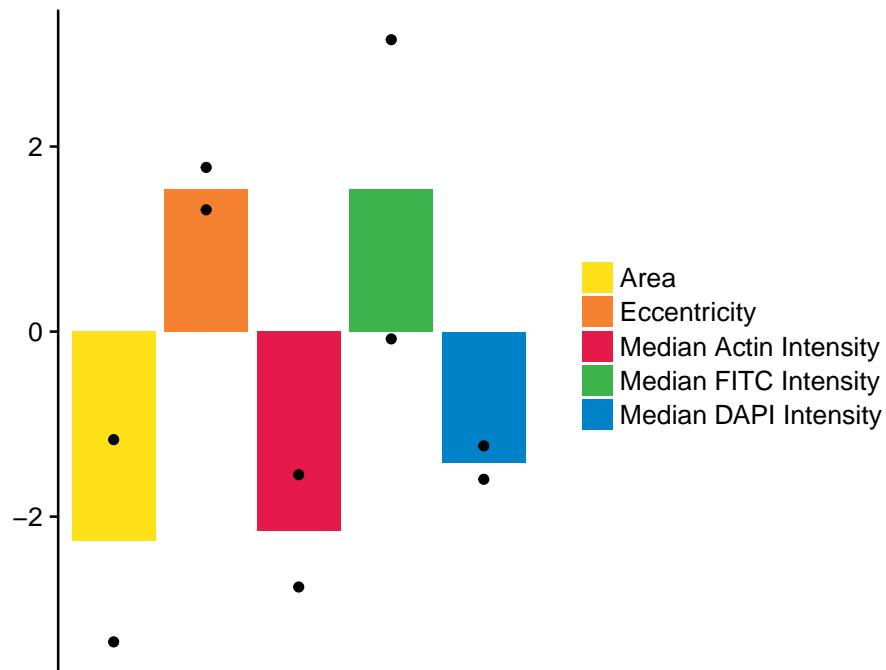
D021T01 MAPK



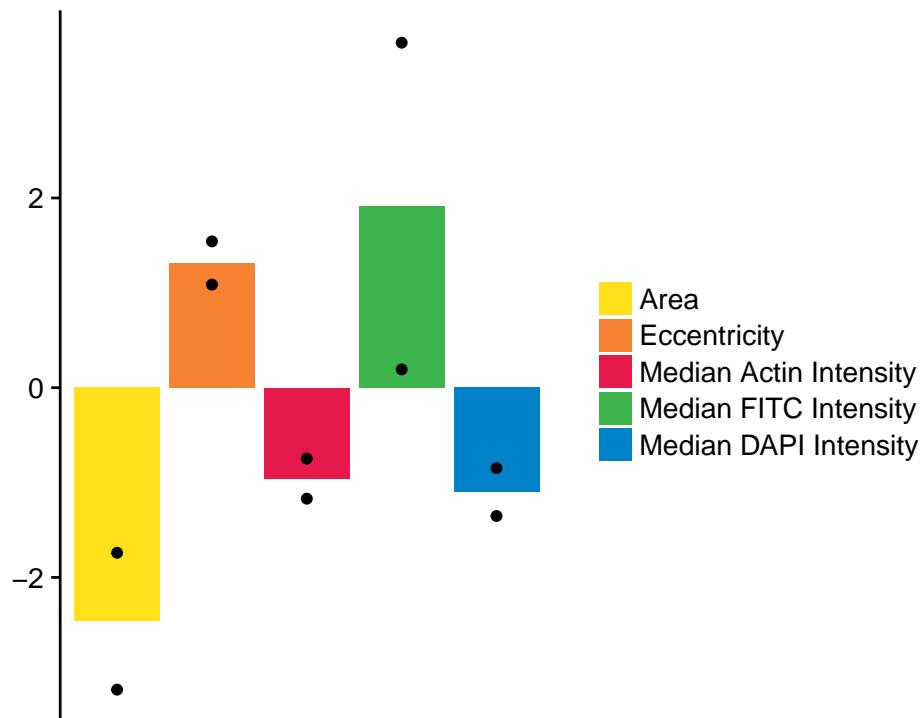
D022T01 Cell Cycle



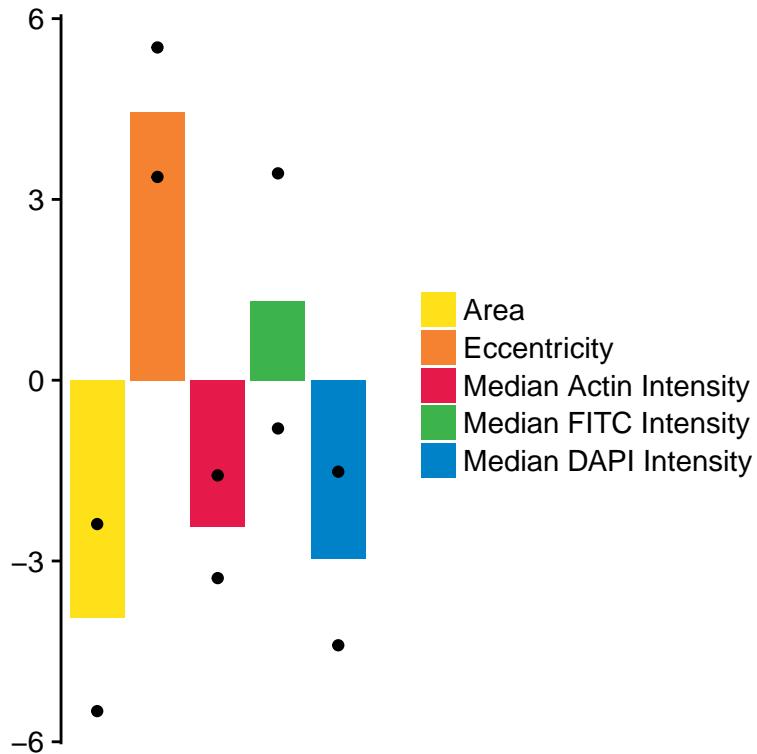
D022T01 Protein Tyrosine Kinase



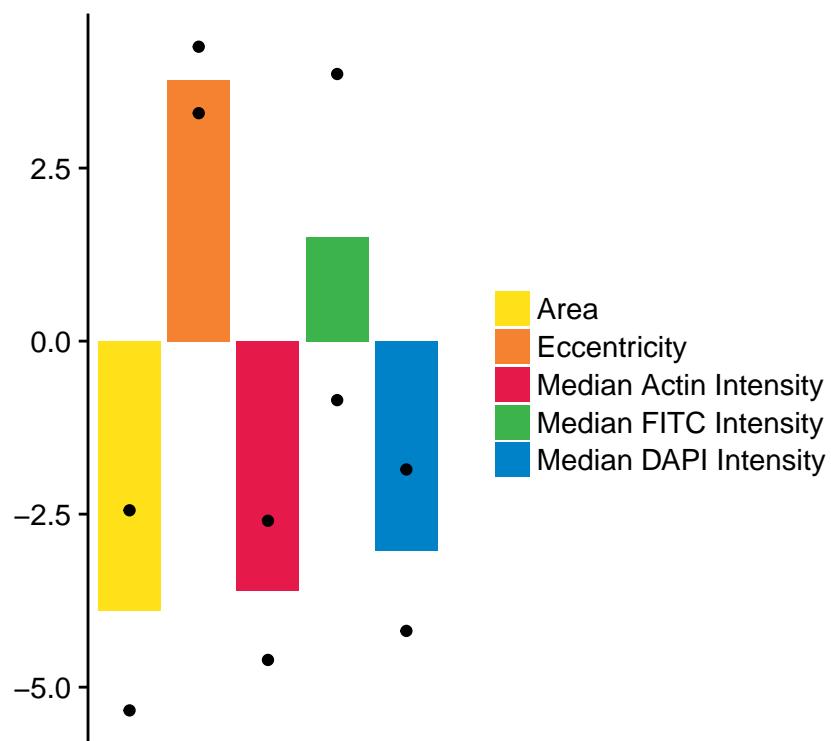
D022T01 PI3K/Akt/mTOR



D022T01 Cytoskeletal Signaling

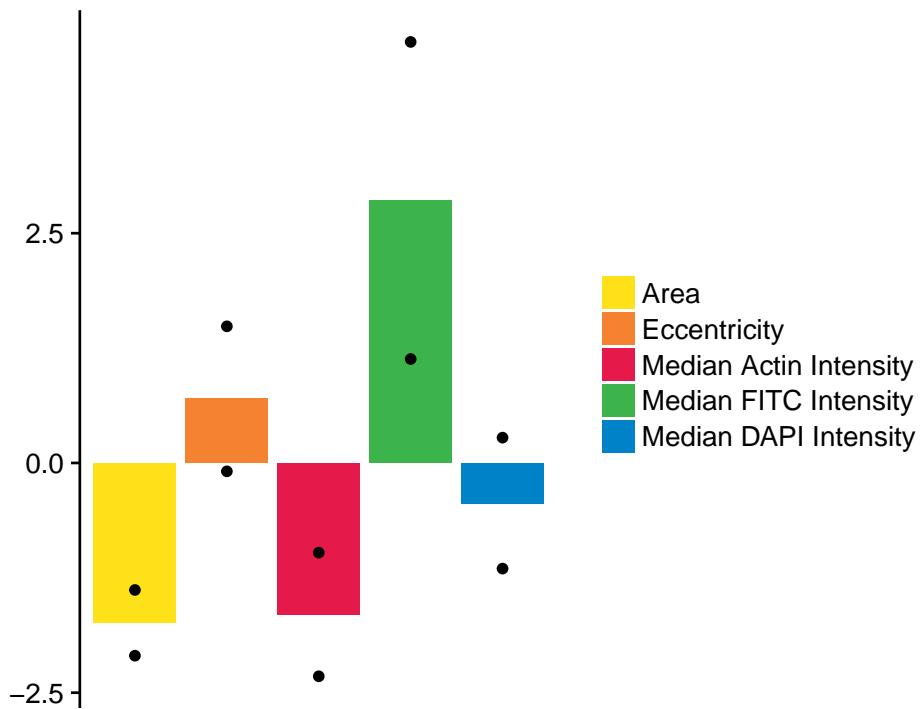


D022T01 Angiogenesis

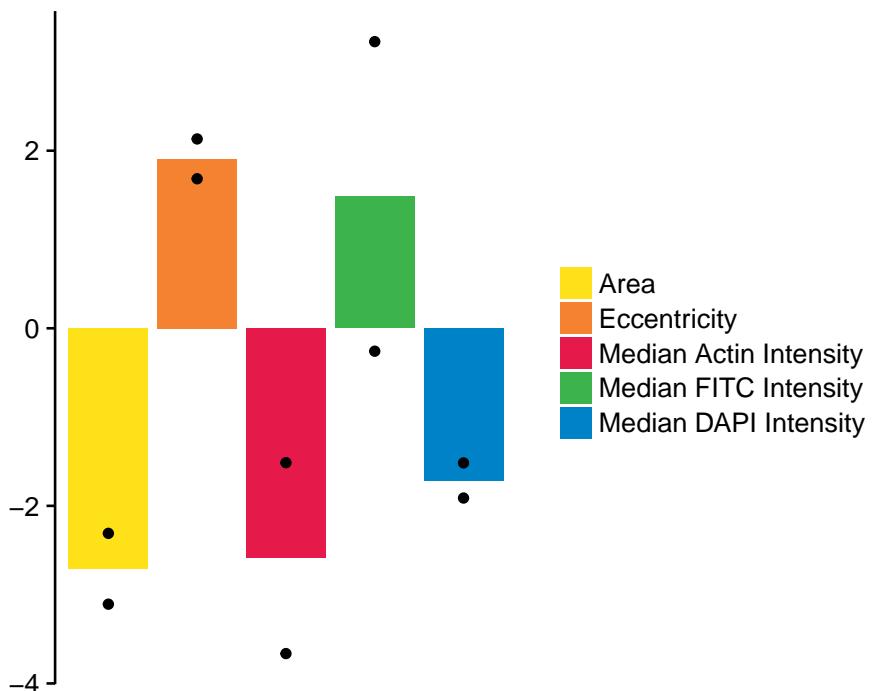


Drug-Induced Phenotypes

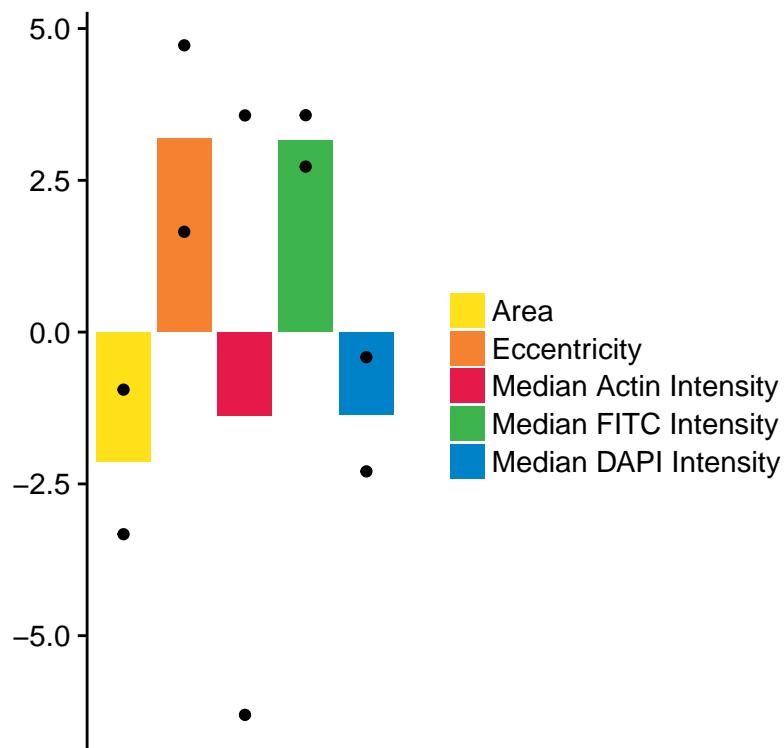
D022T01 MAPK



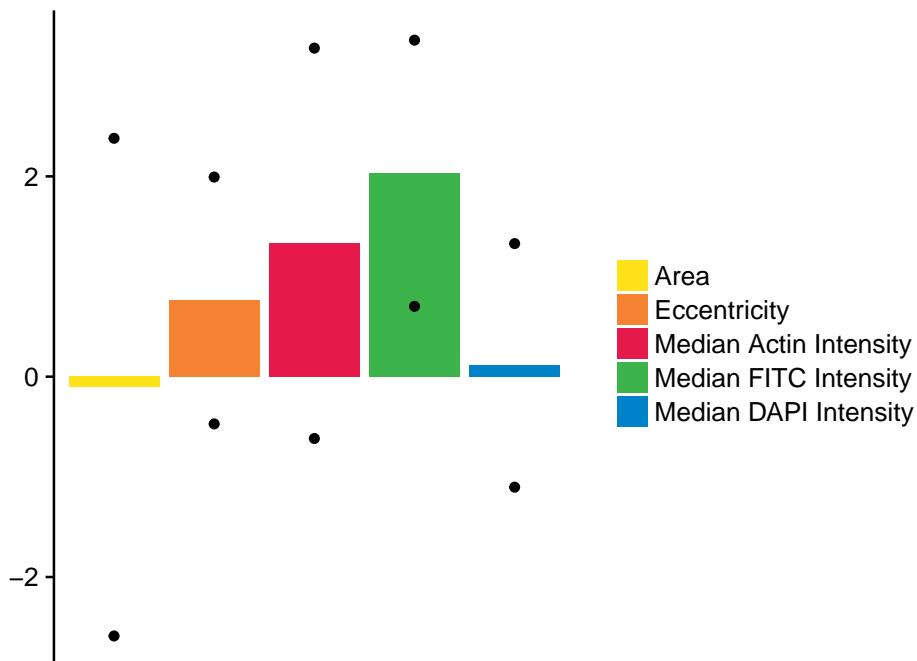
D027T01 Protein Tyrosine Kinase



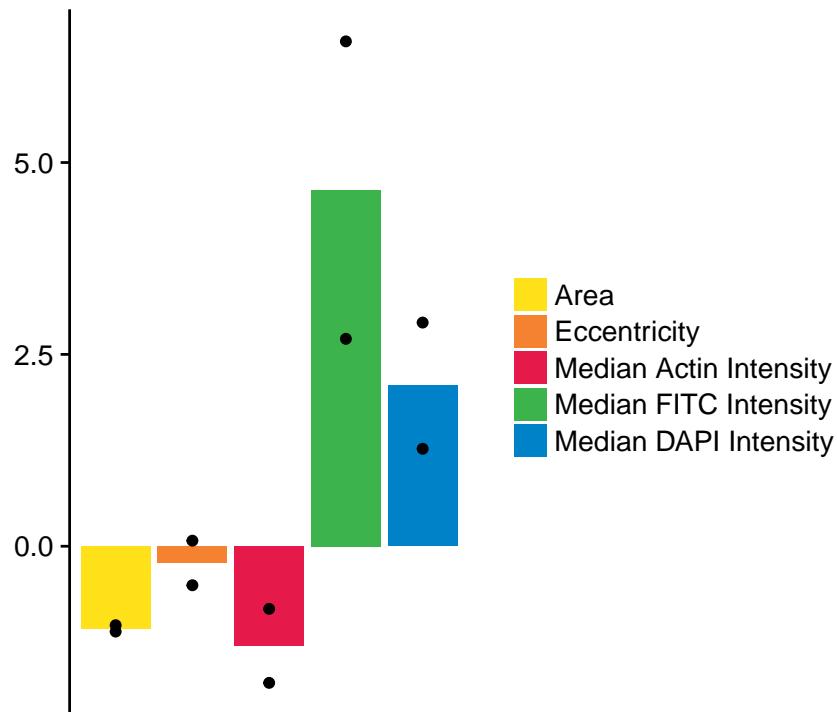
D027T01 PI3K/Akt/mTOR



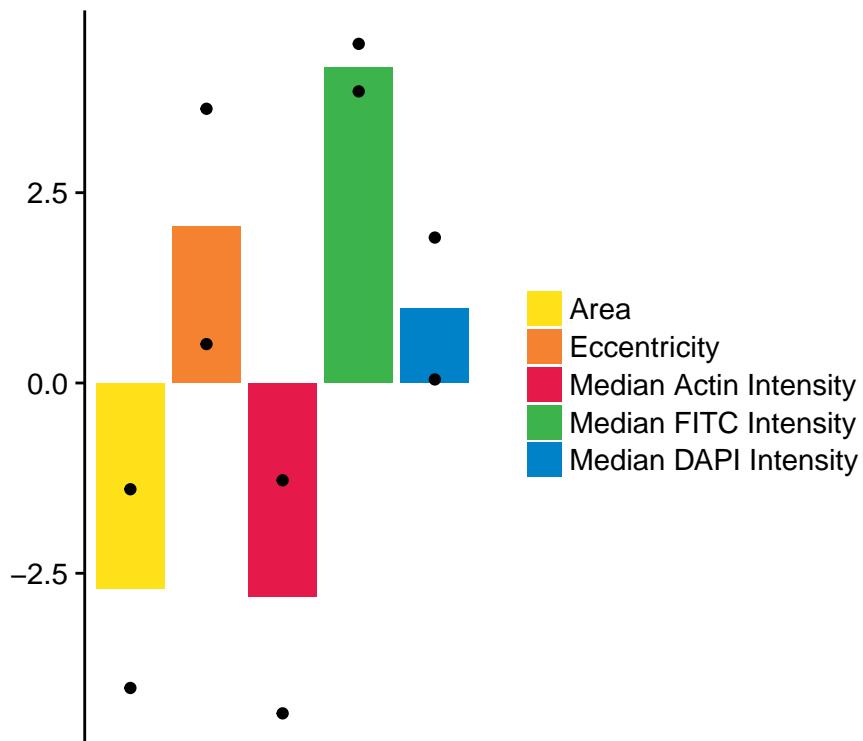
D027T01 Cell Cycle



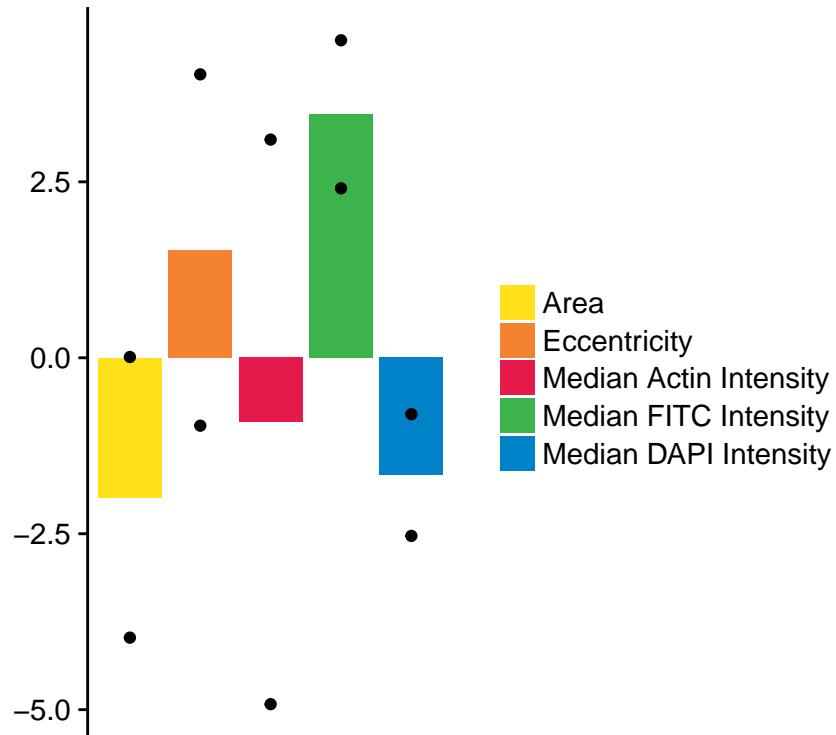
D030T01 Protein Tyrosine Kinase



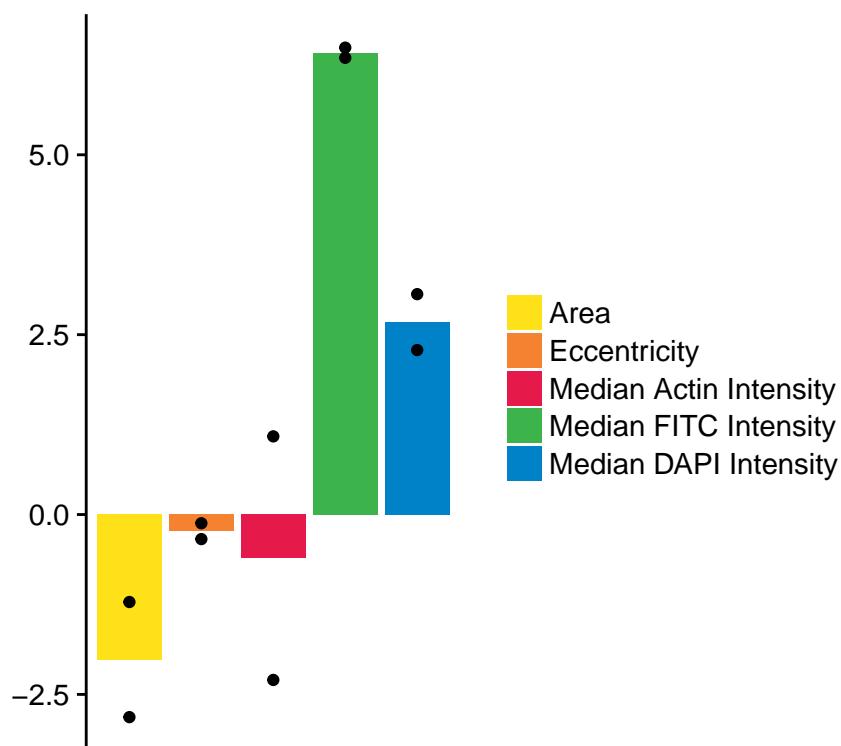
D030T01 PI3K/Akt/mTOR



D030T01 Cell Cycle

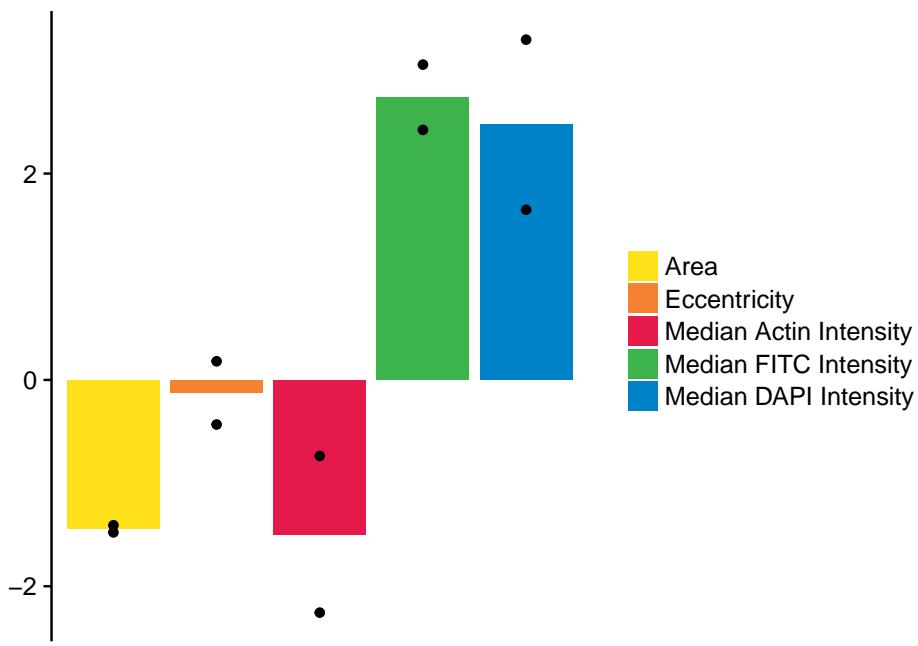


D030T01 MAPK

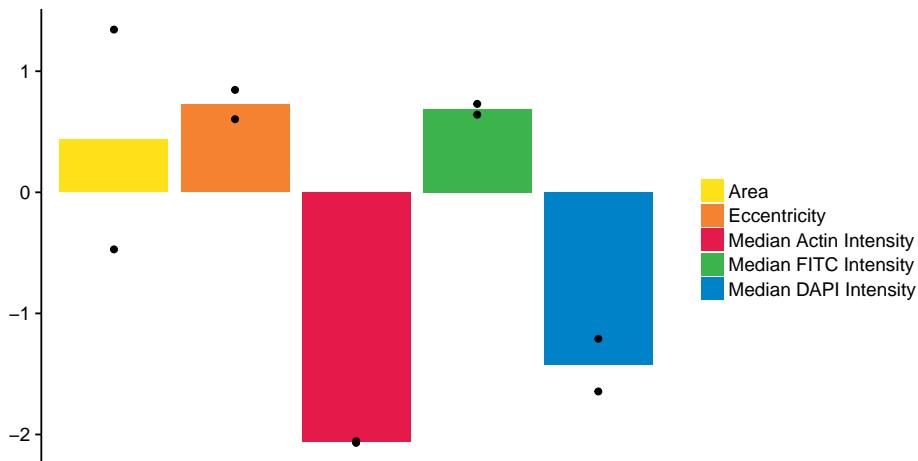


Drug-Induced Phenotypes

D030T01 Angiogenesis

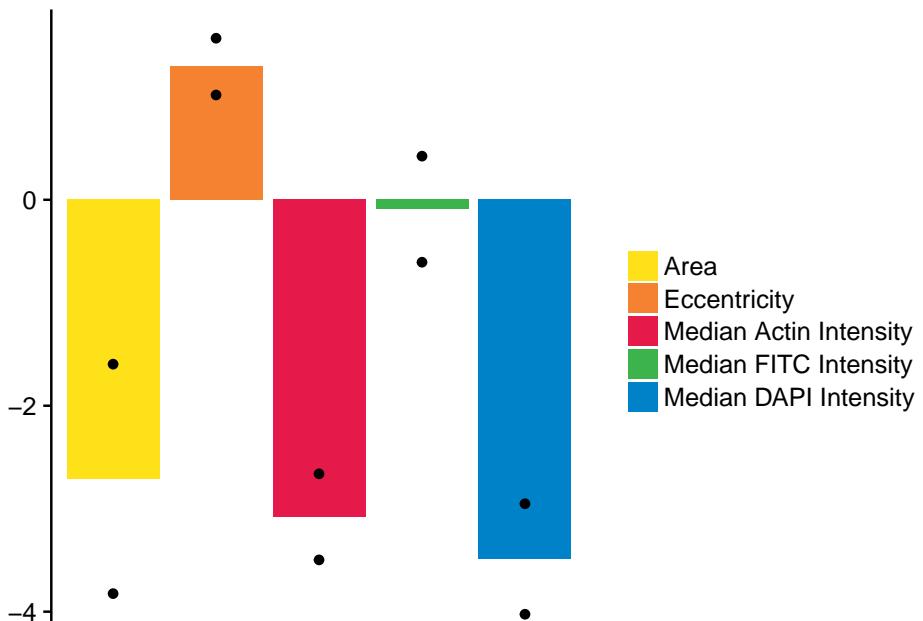


D046T01 PI3K/Akt/mTOR

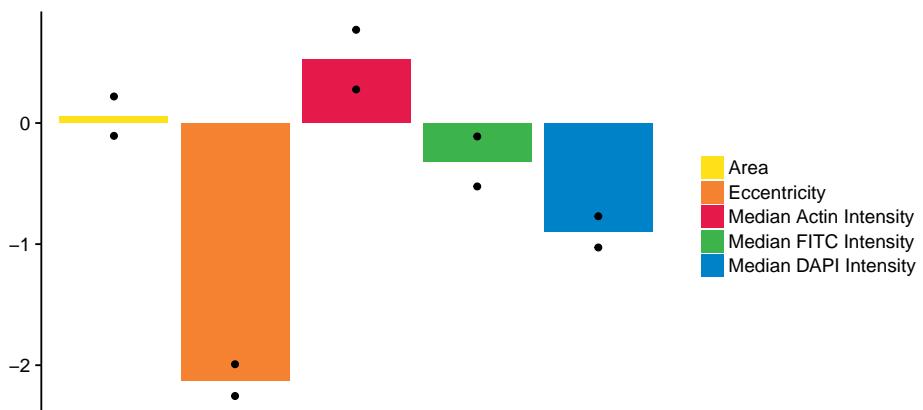


Drug-Induced Phenotypes

D046T01 Cell Cycle

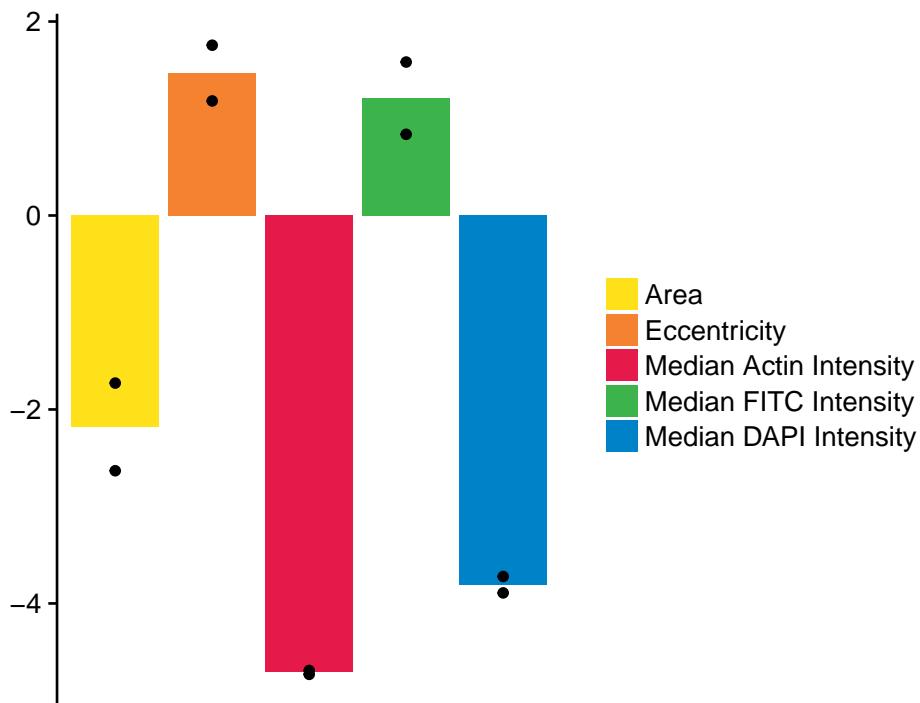


D046T01 Angiogenesis

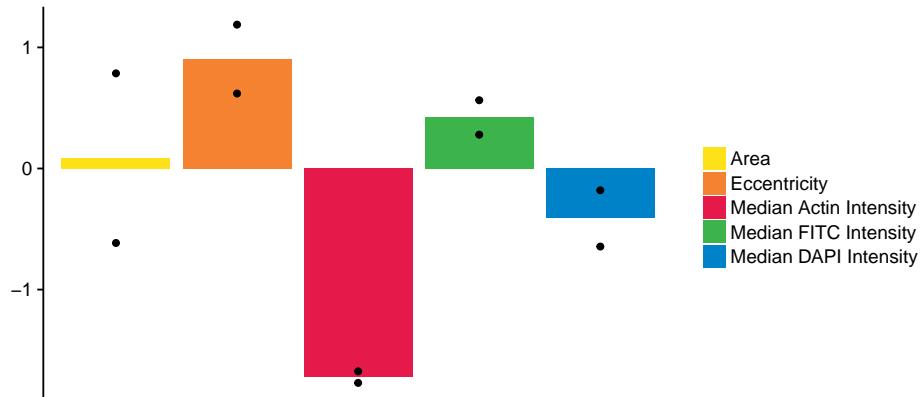


Drug-Induced Phenotypes

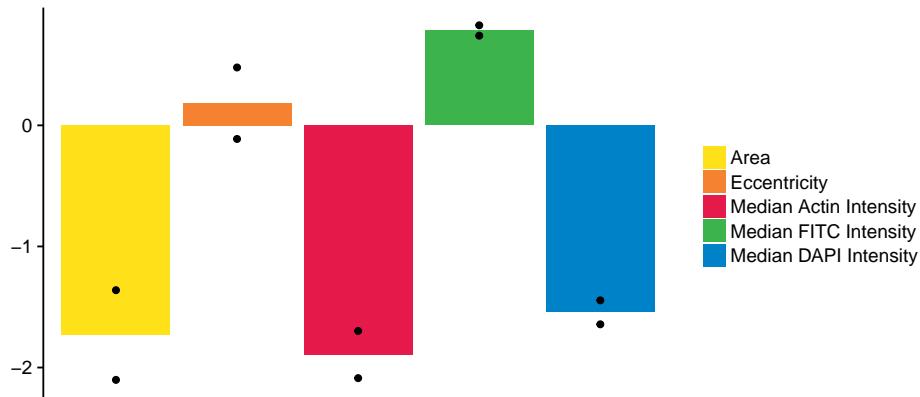
D046T01 MAPK



D046T01 TGF-beta/Smad

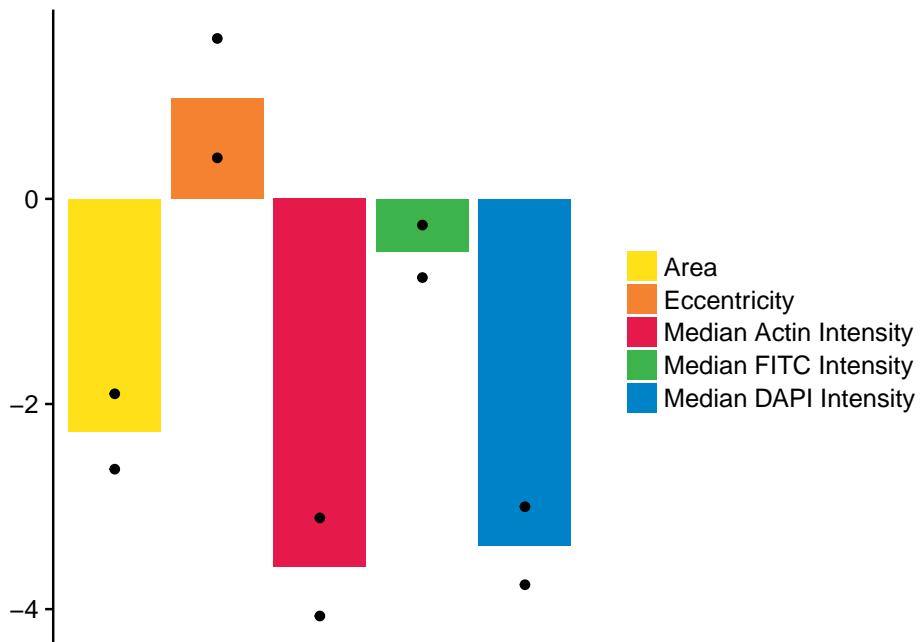


D055T01 PI3K/Akt/mTOR

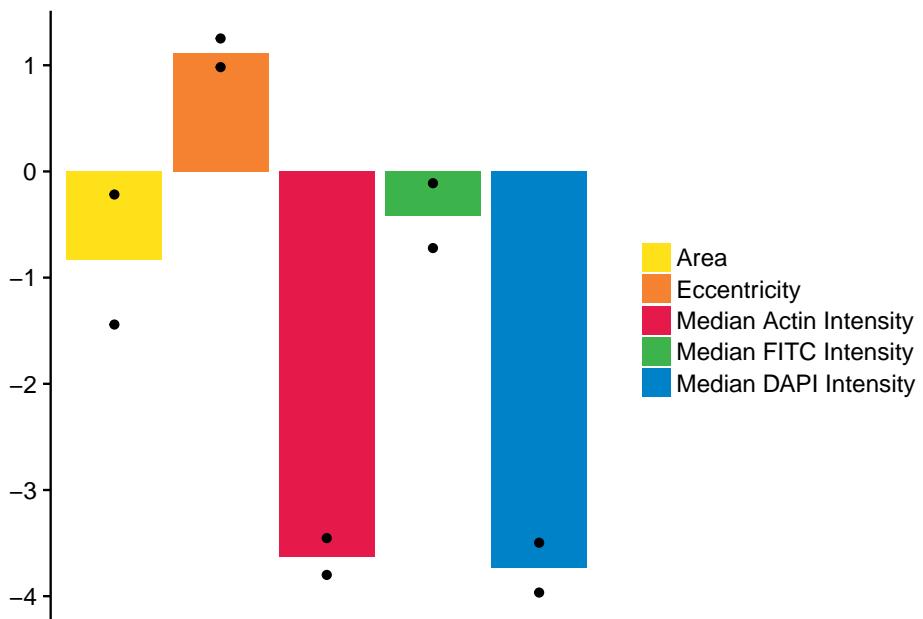


Drug-Induced Phenotypes

D055T01 Cell Cycle

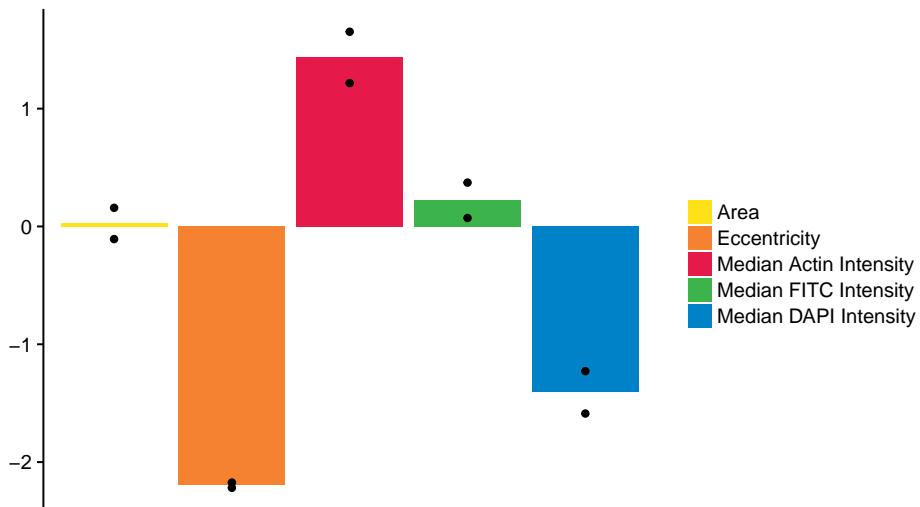


D055T01 MAPK

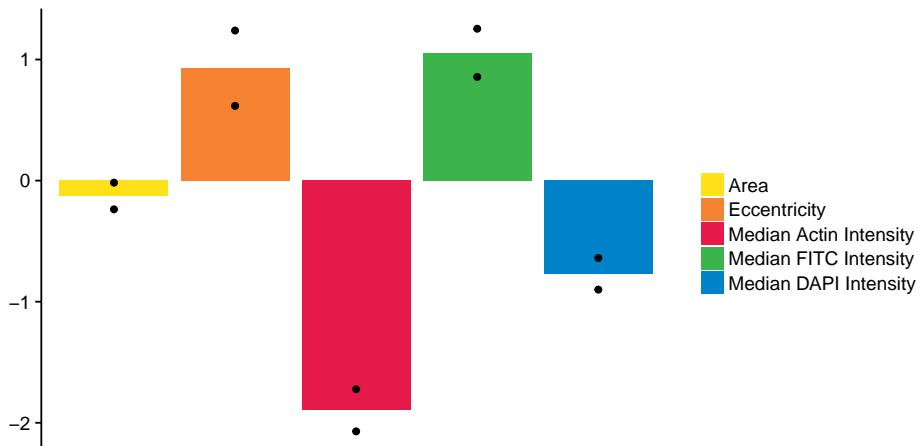


Drug-Induced Phenotypes

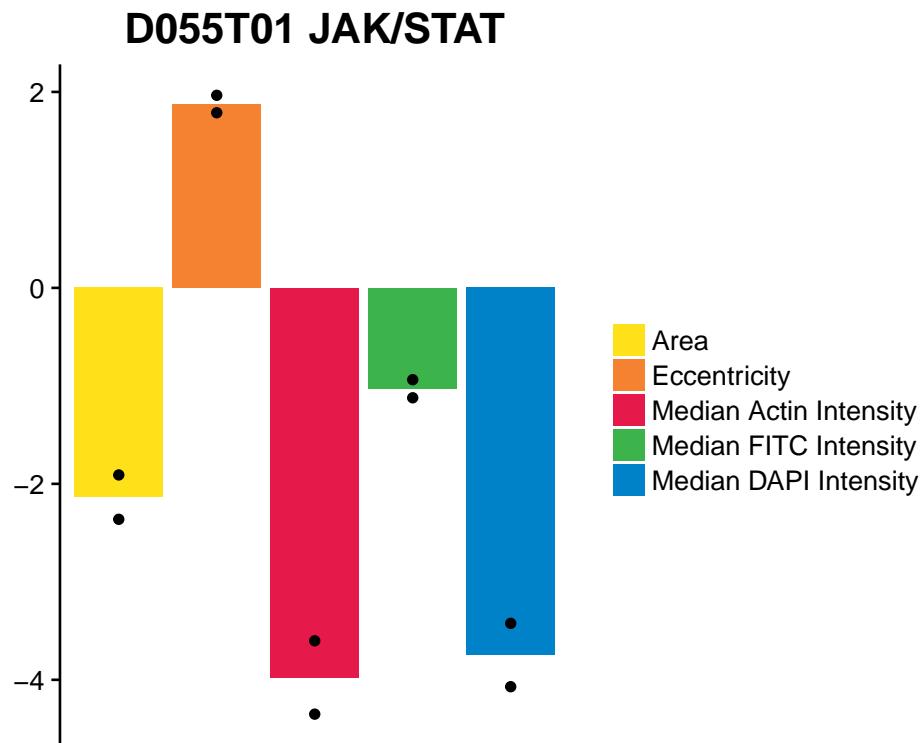
D055T01 Angiogenesis



D055T01 TGF-beta/Smad



Drug-Induced Phenotypes



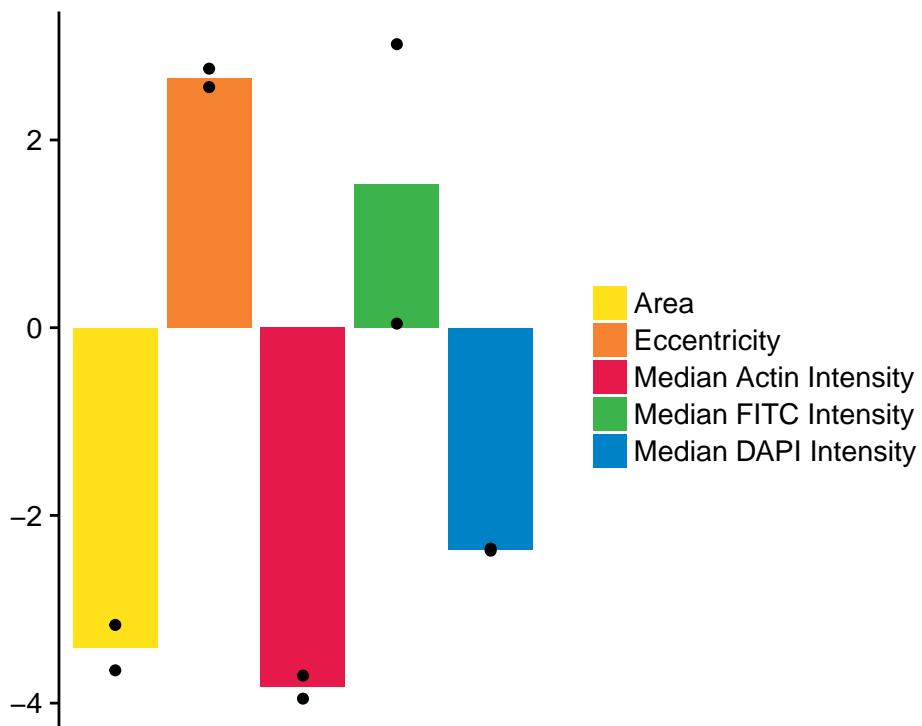
3.3.2 Phenoprints for Targets

```
for(line in unique(all_target_features$Line)) {  
  labels = unique(all_target_features[  
    all_target_features$Line == line, "Label"])  
  for(label in labels) {  
    ggplot_df = all_target_features[all_target_features$Line == line &  
      all_target_features$Label == label, ]  
    ggplot_df$Line = NULL  
    ggplot_df$Label = NULL  
    ggplot_df = melt(ggplot_df, id.vars = c("Replicate"))  
    ggplot_df$variable = key_features[ggplot_df$variable]  
    ggplot_df$variable = factor(ggplot_df$variable, levels = key_features)  
    ggplot_df_bar = aggregate(  
      x = ggplot_df$value,  
      by = list("variable" = ggplot_df$variable),  
      FUN = median)  
    colnames(ggplot_df_bar) = c("variable", "value")  
    gp = ggplot(mapping = aes(x = variable, y = value)) +  
      geom_col(data = ggplot_df_bar, mapping = aes(fill = variable)) +  
      geom_point(data = ggplot_df) +  
      scale_fill_manual(values = pheno_color_scale) +  
      theme_vignette() + coord_fixed() +  
      theme(  
        axis.text.x = element_blank(), axis.line.x = element_blank(),  
        axis.ticks.x = element_blank(), legend.title = element_blank()) +
```

Drug-Induced Phenotypes

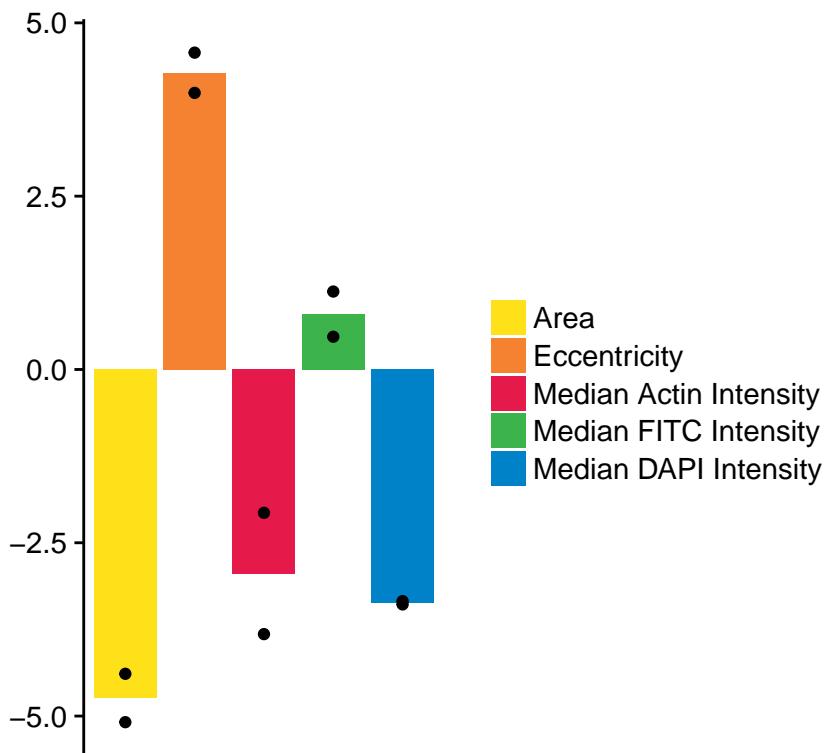
```
    xlab("") + ylab("") + ggtitle(paste0(line, " ", label))
    plot(gp)
    if(save_images) ggsave(
      filename = file.path(img_out_dir, sprintf(
        "Phenoprints_Tests_%s_%s.pdf", line, make.names(label))),
      width = 5, height = 5, useDingbats = FALSE)
  }
}
```

D004T01 aurora kinase

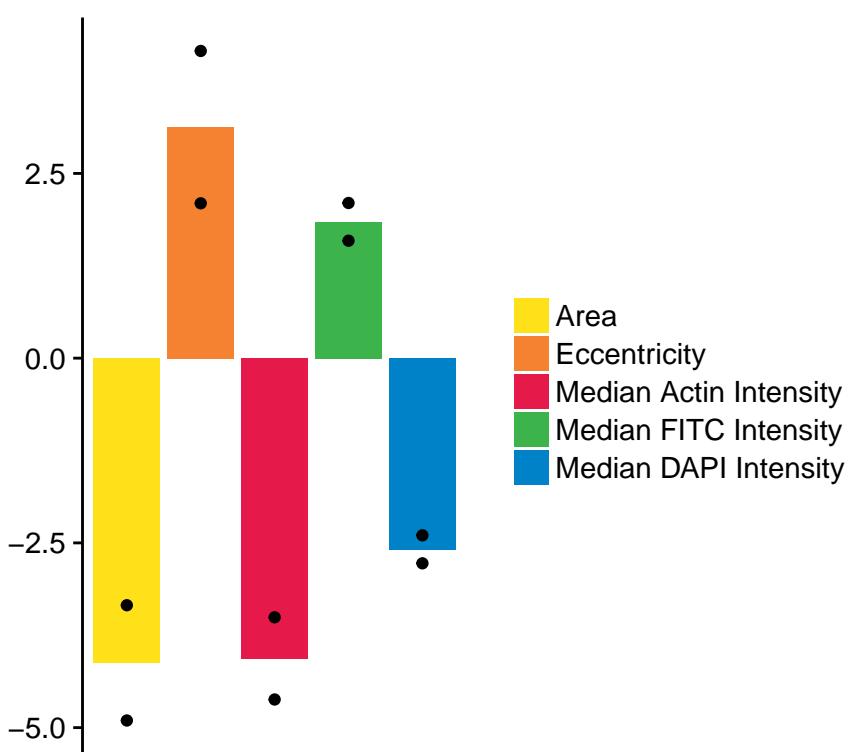


Drug-Induced Phenotypes

D004T01 alk

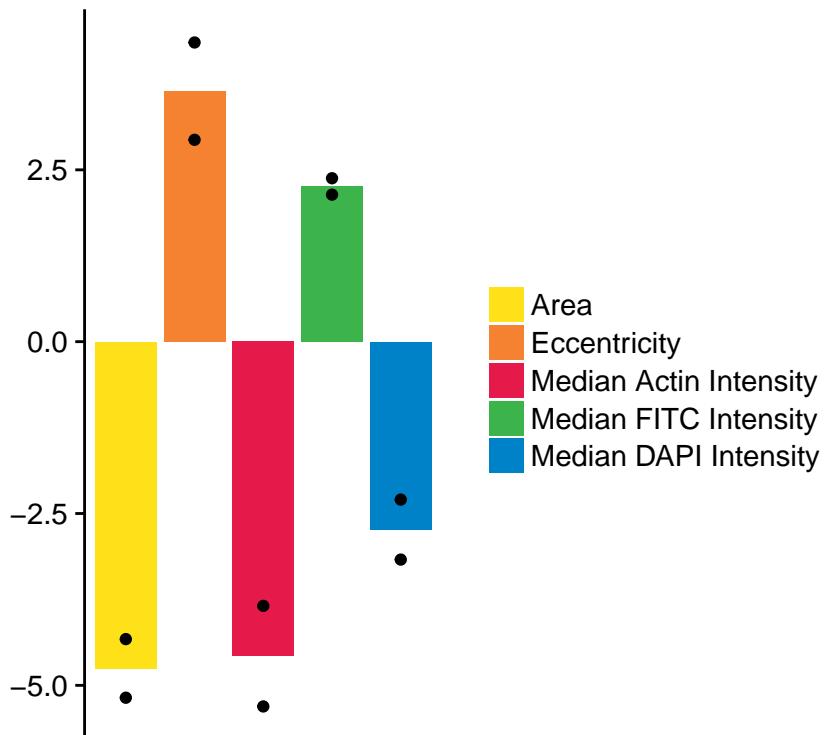


D004T01 cdk

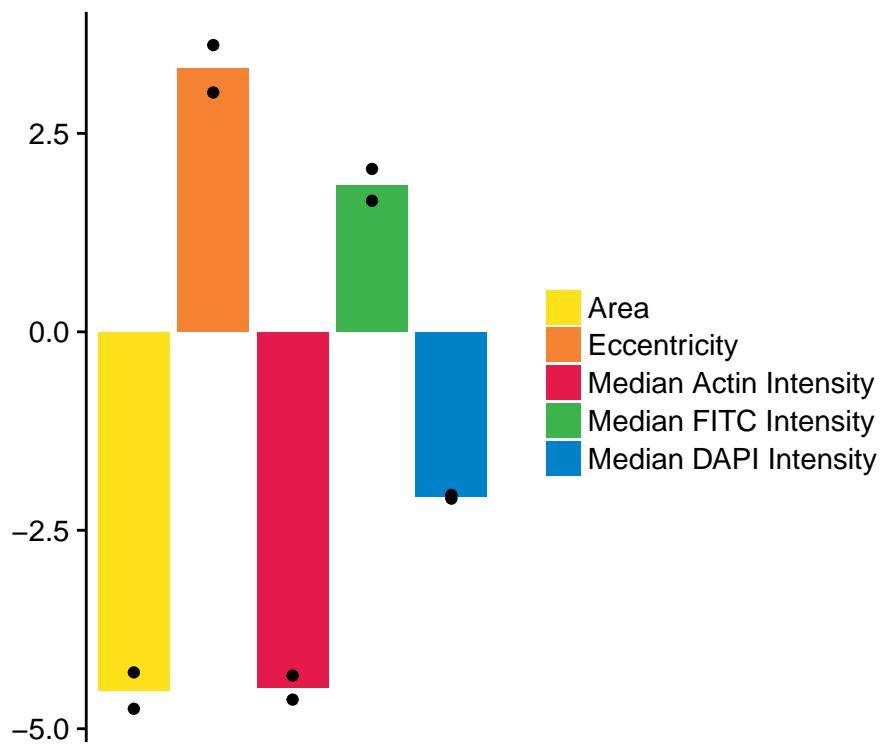


Drug-Induced Phenotypes

D004T01 jak

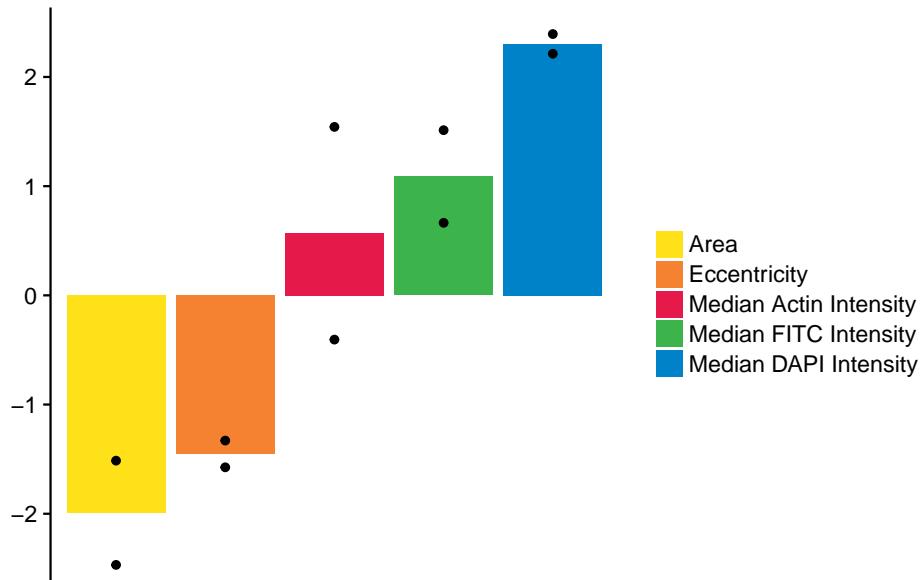


D004T01 gsk-3

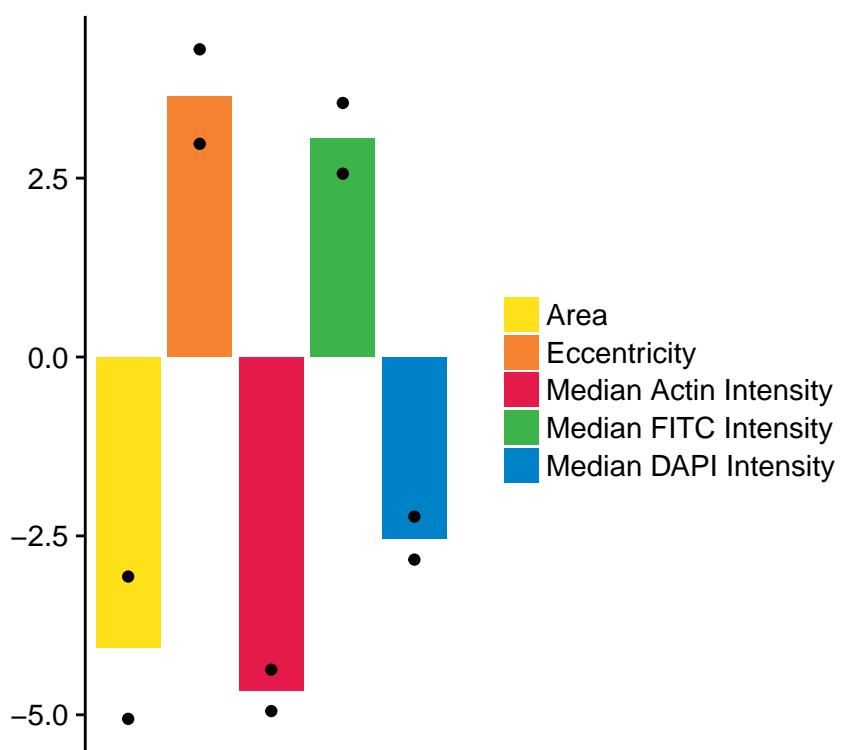


Drug-Induced Phenotypes

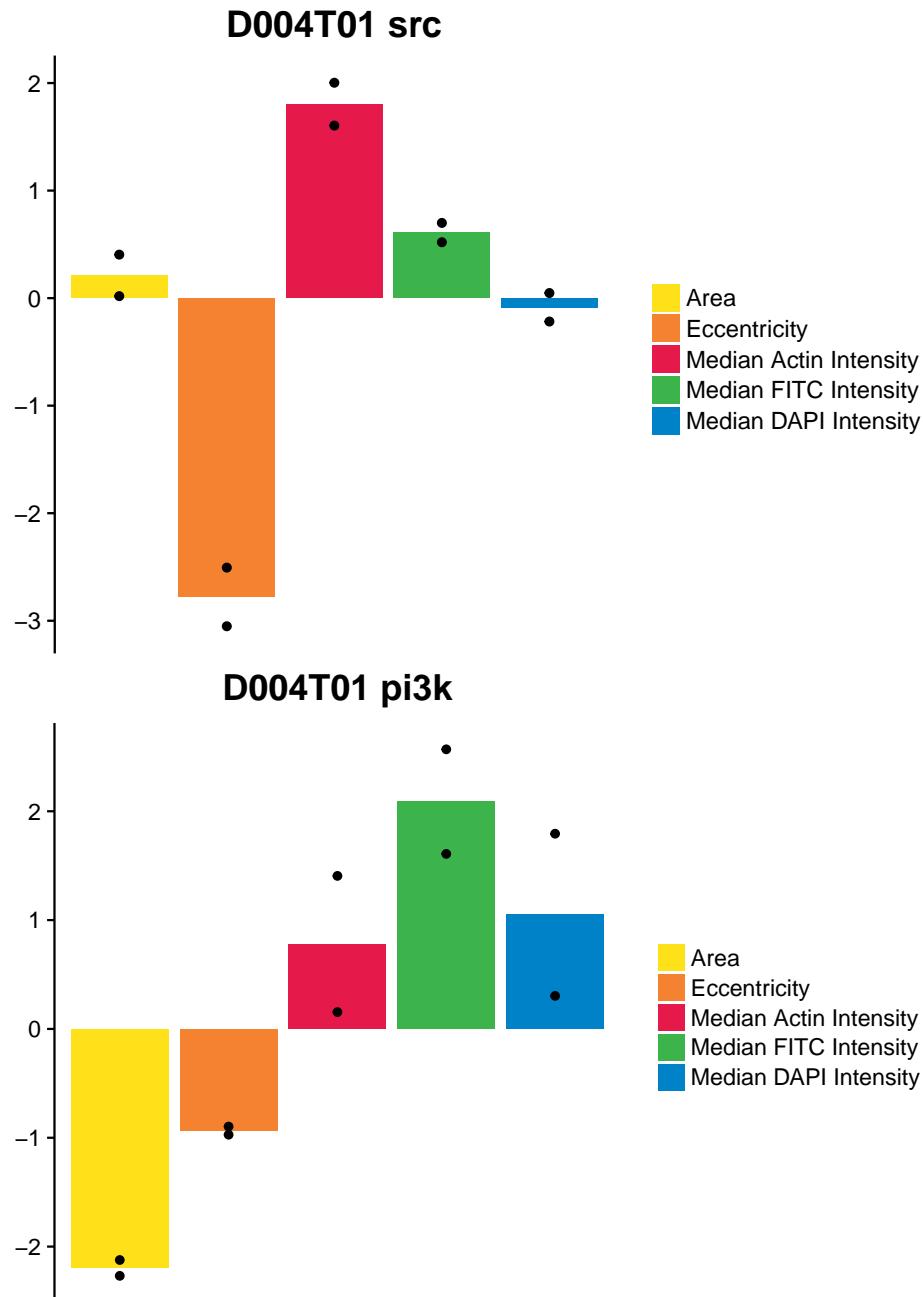
D004T01 mtor



D004T01 mek

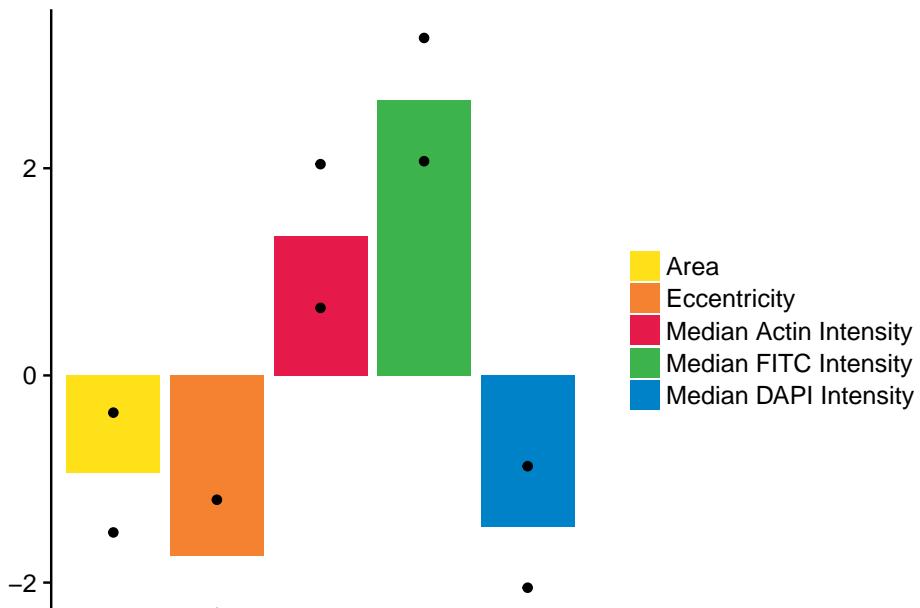


Drug-Induced Phenotypes

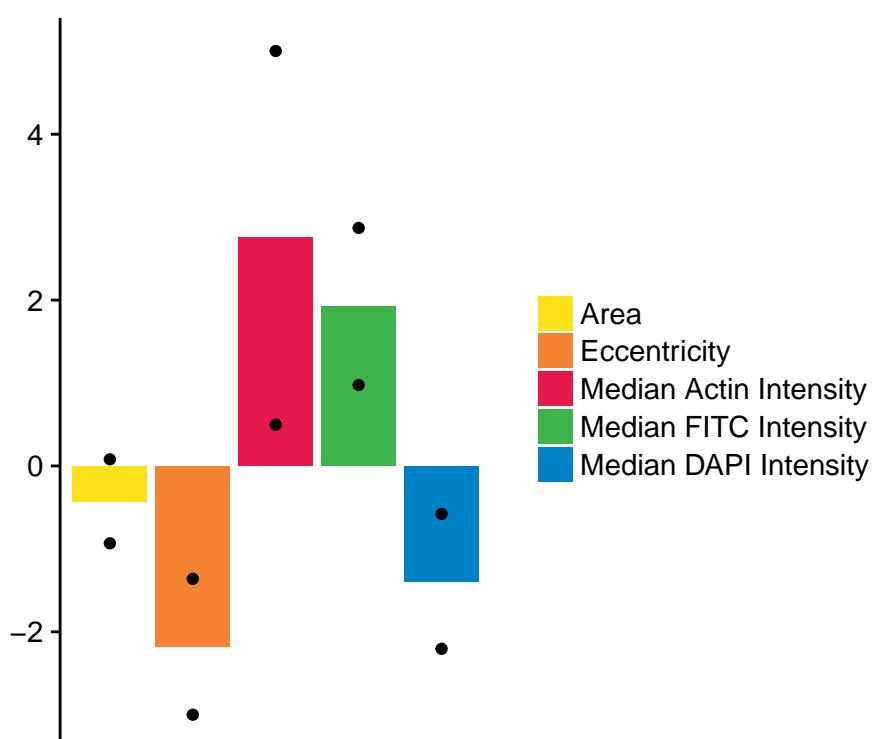


Drug-Induced Phenotypes

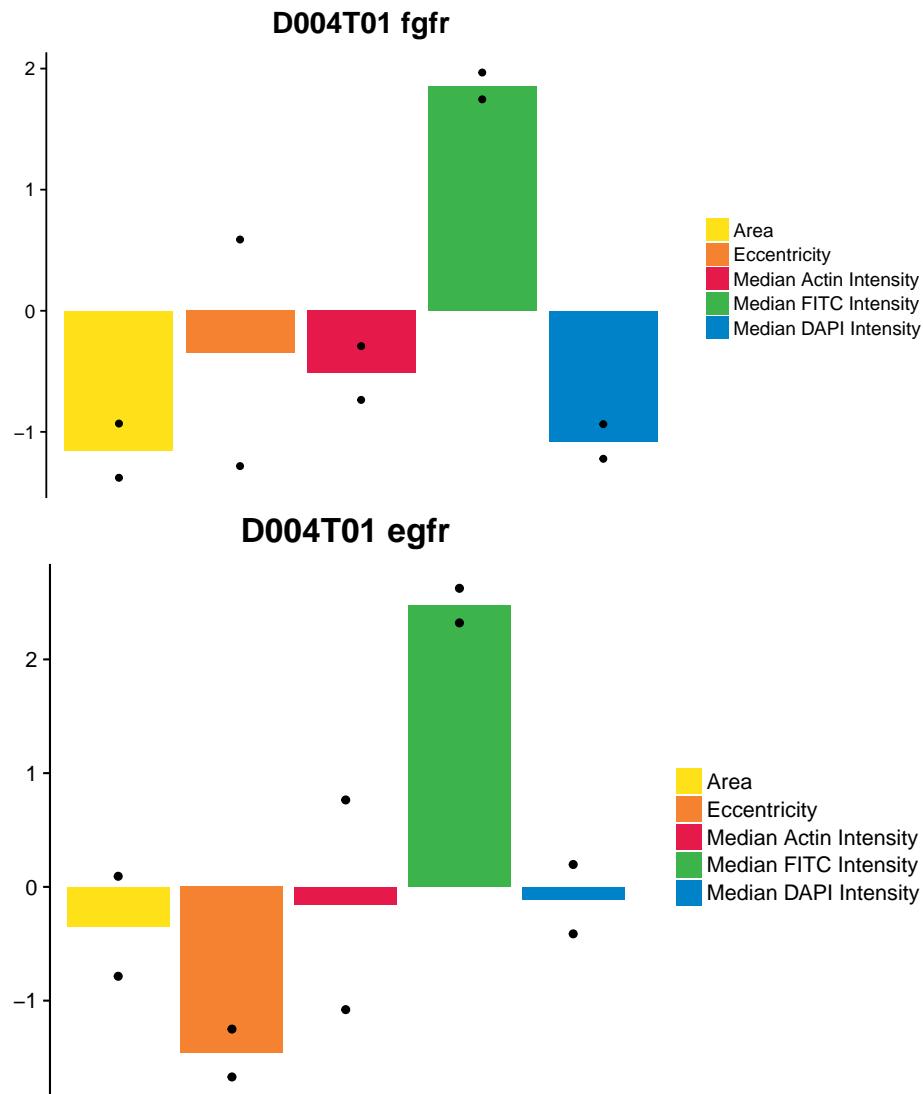
D004T01 *flt*



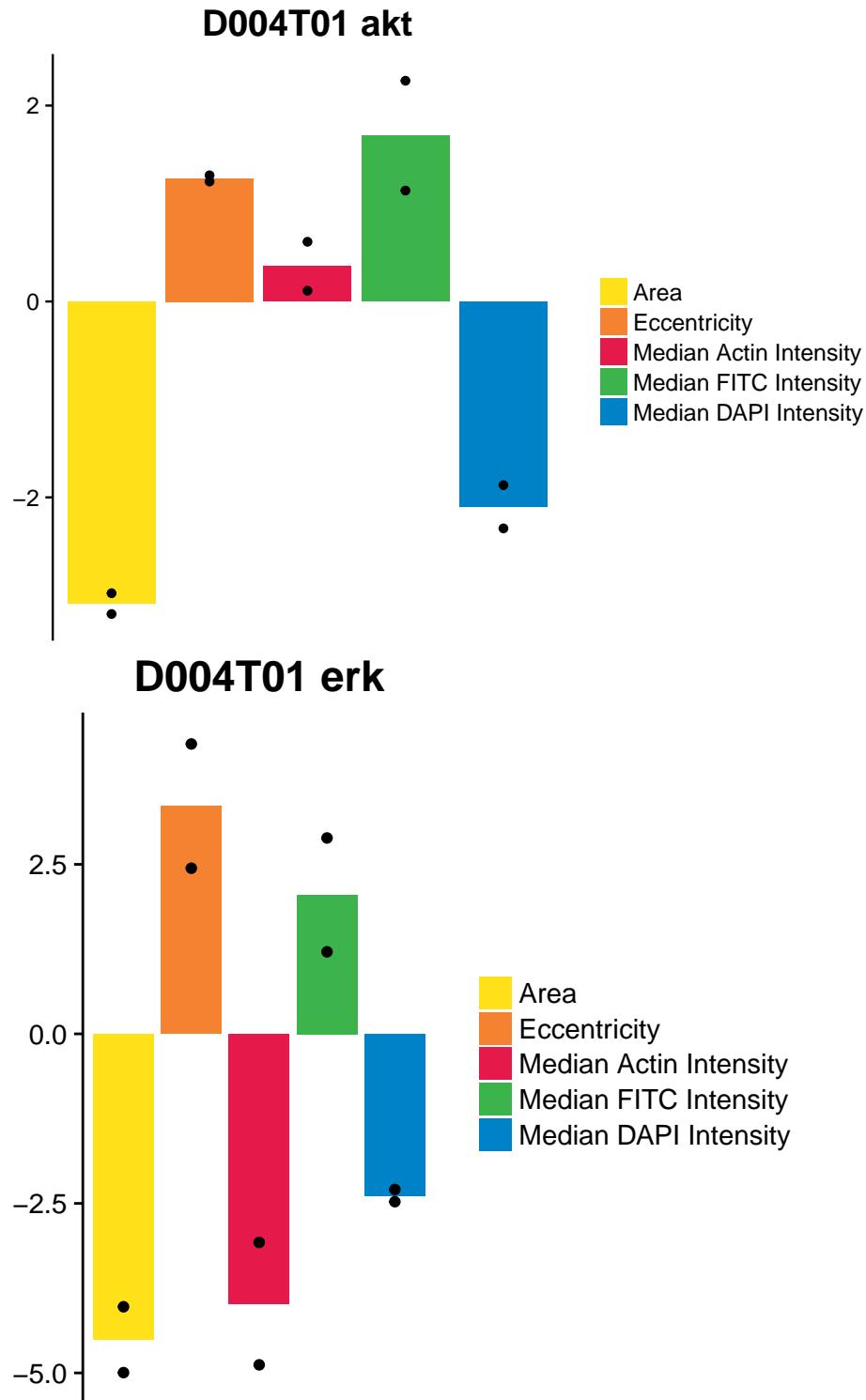
D004T01 *pdgfr*



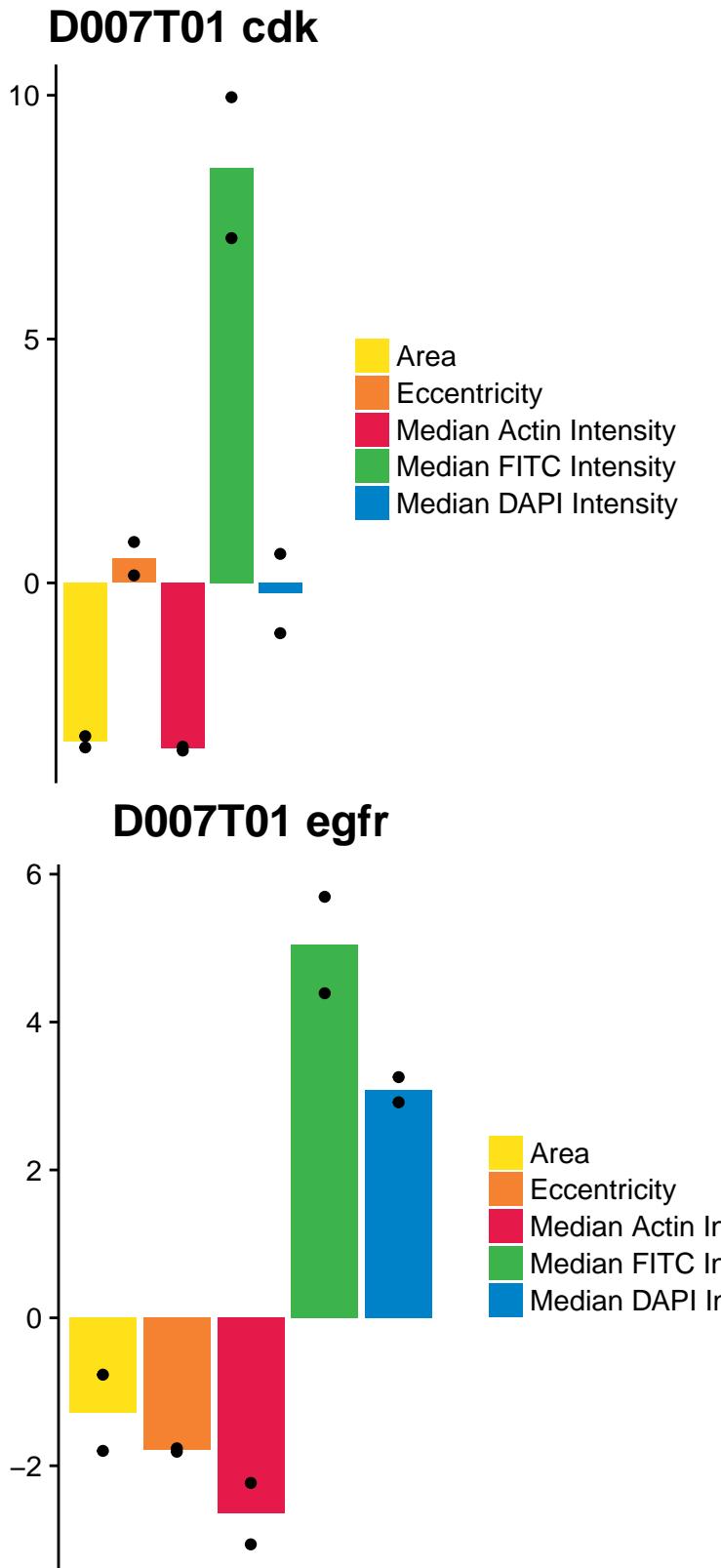
Drug-Induced Phenotypes



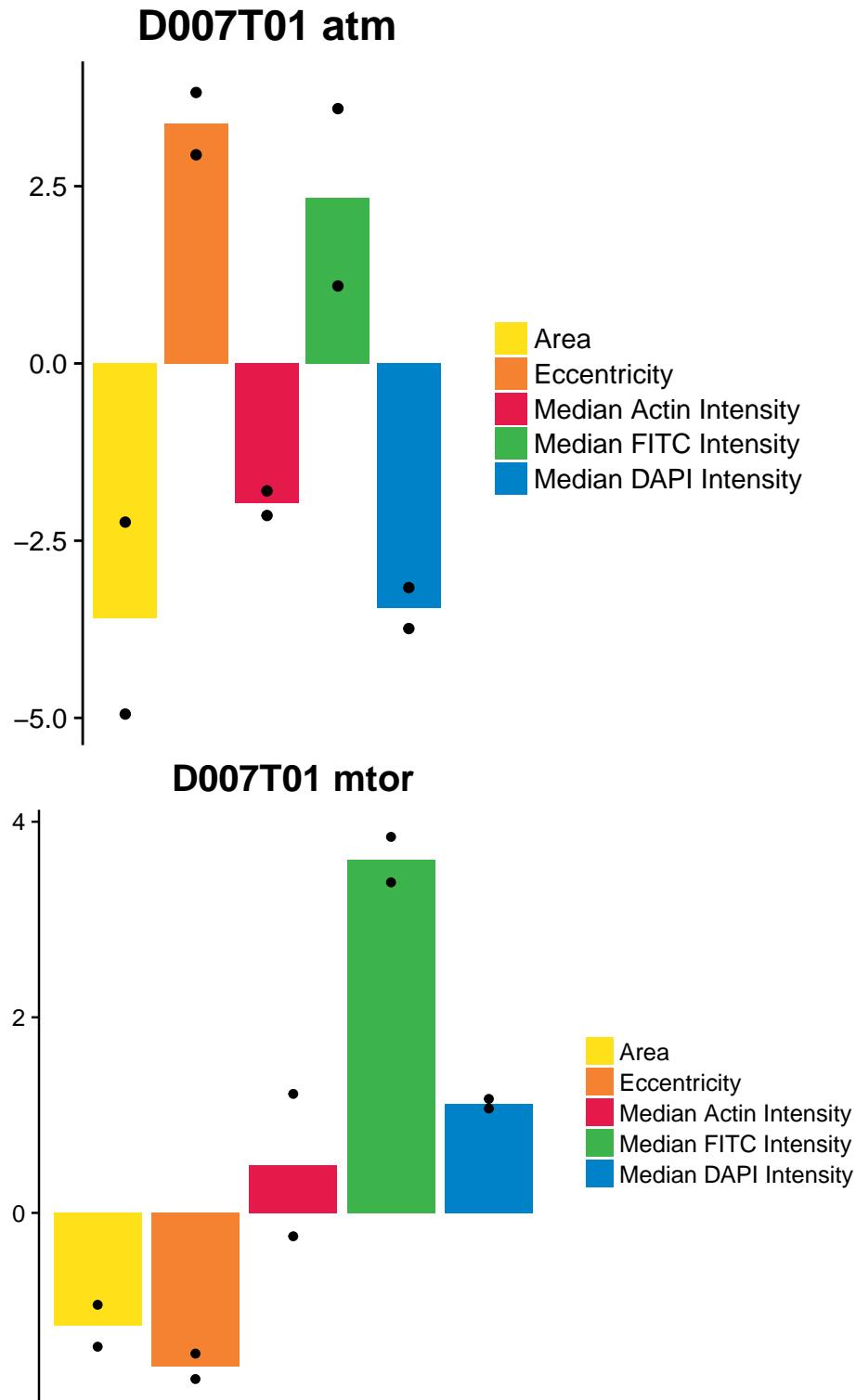
Drug-Induced Phenotypes



Drug-Induced Phenotypes

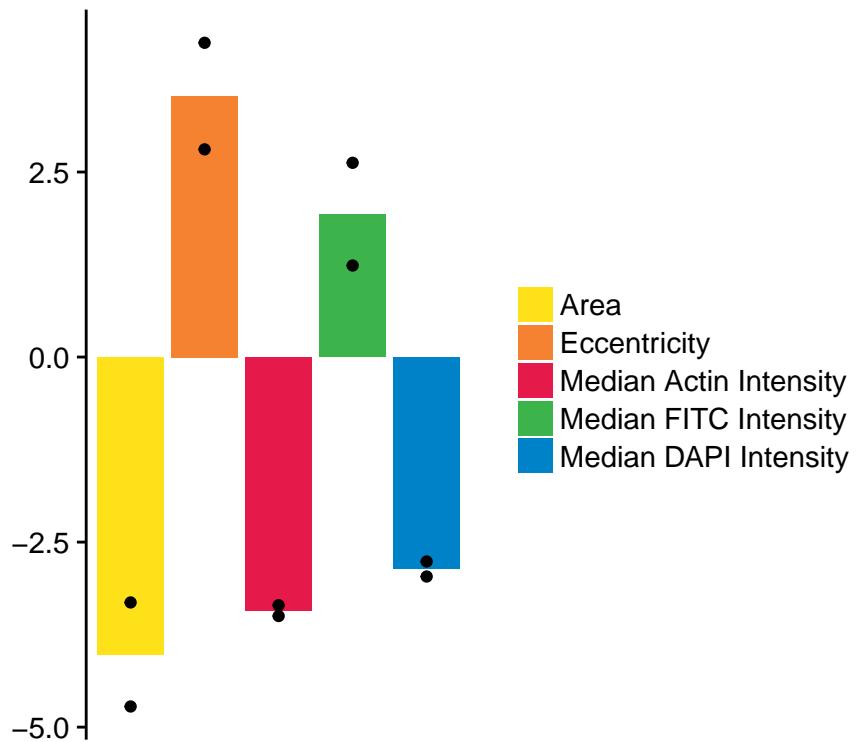


Drug-Induced Phenotypes

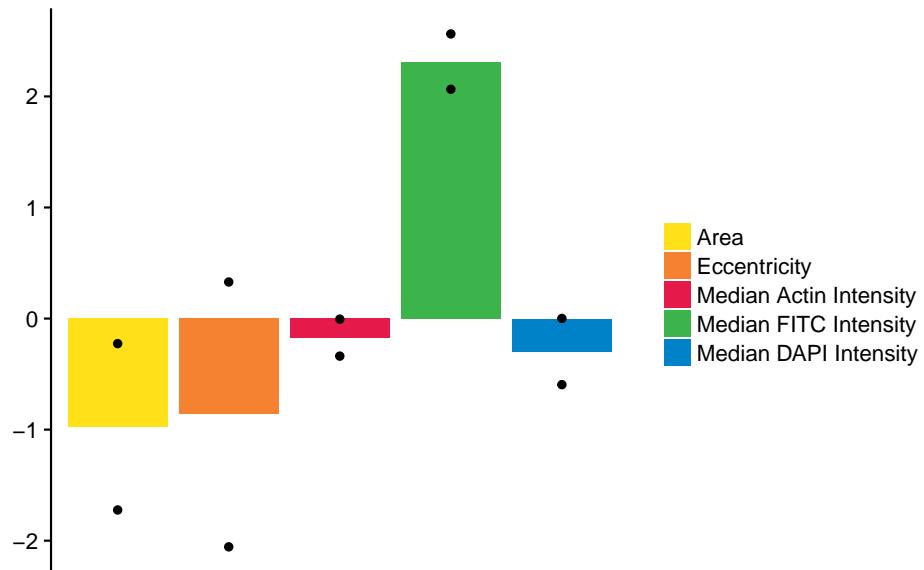


Drug-Induced Phenotypes

D007T01 gsk-3

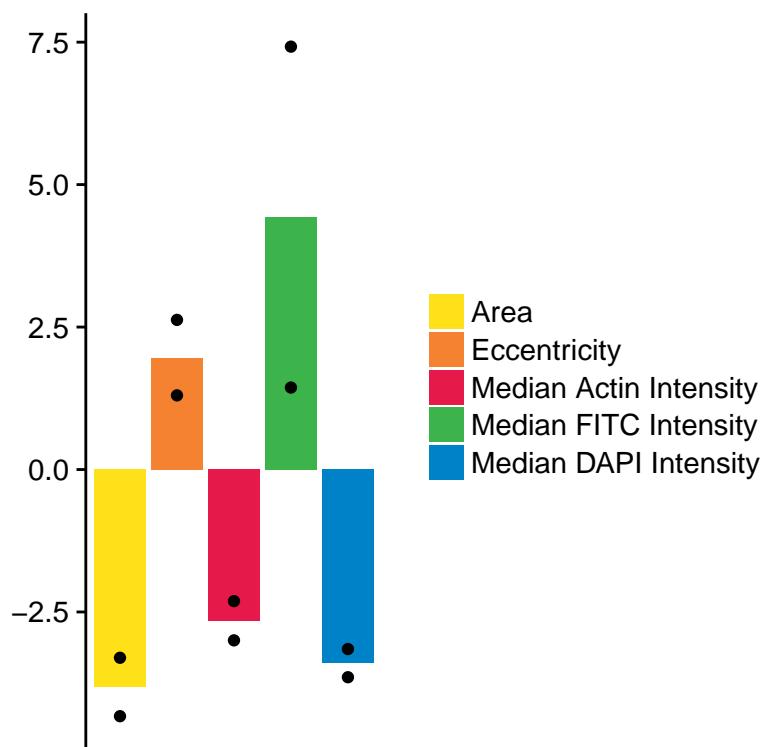


D007T01 akt

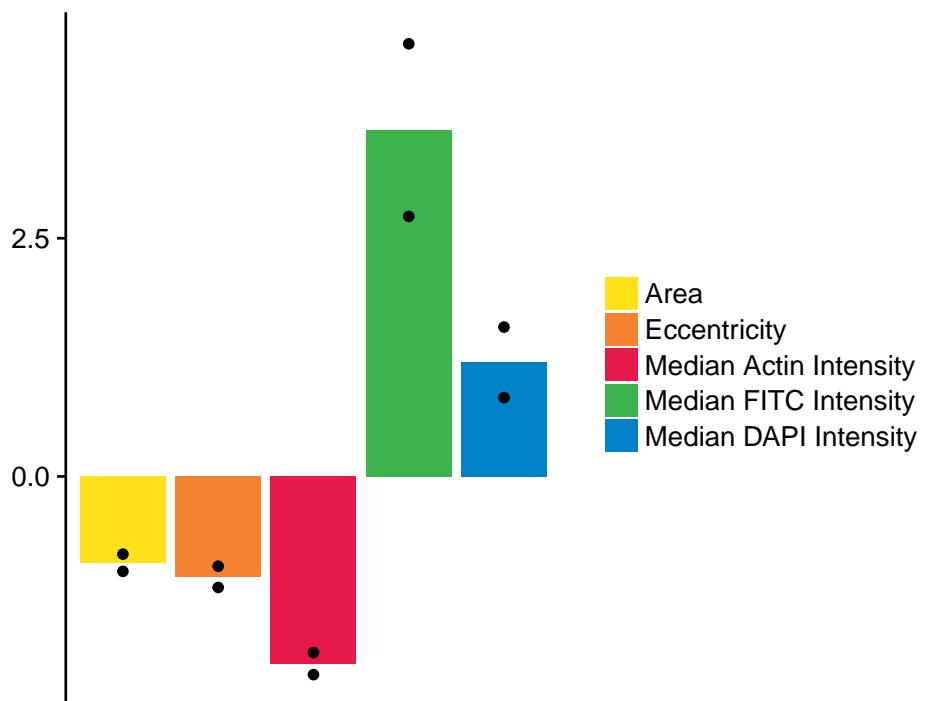


Drug-Induced Phenotypes

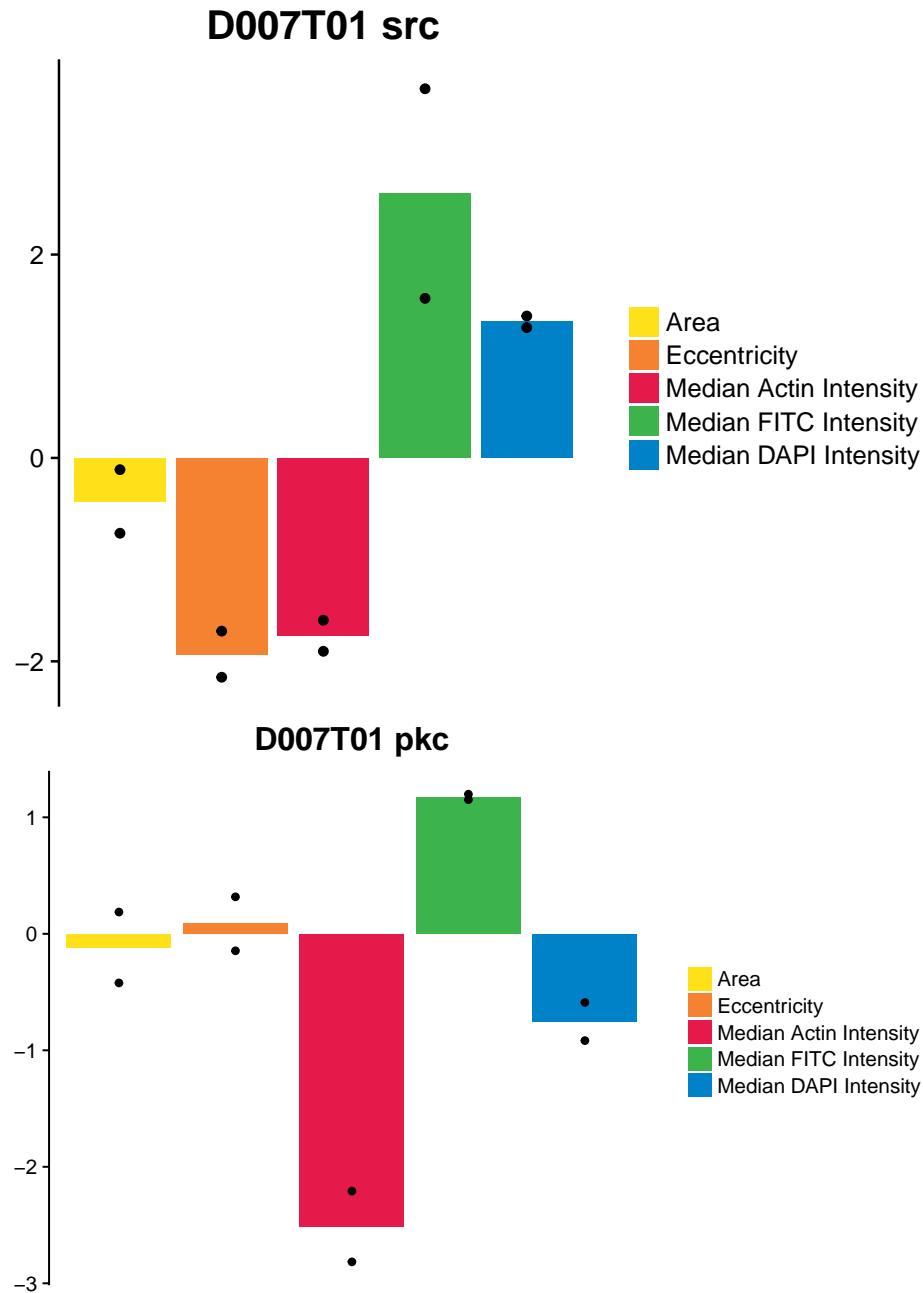
D007T01 chk



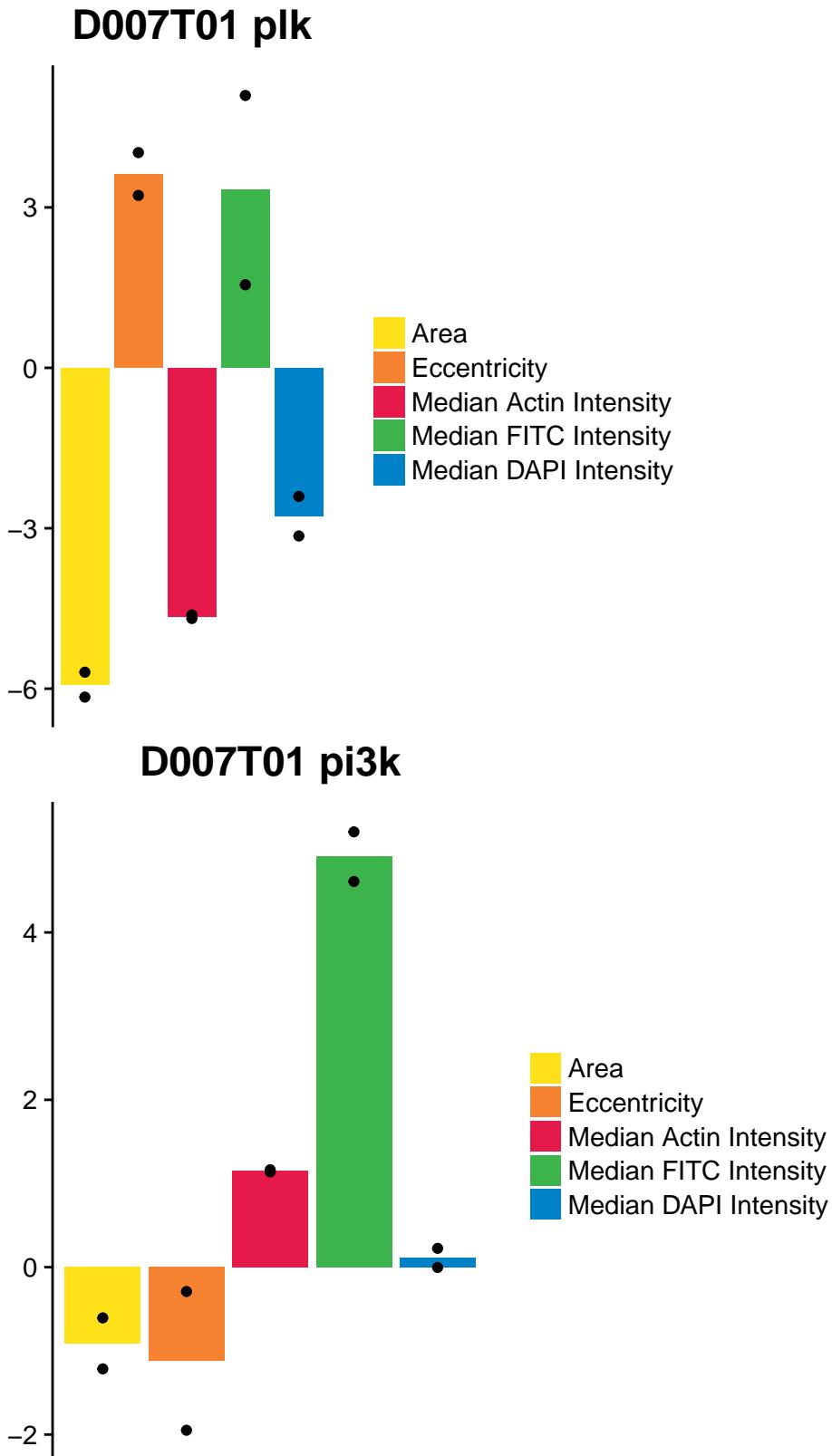
D007T01 mek



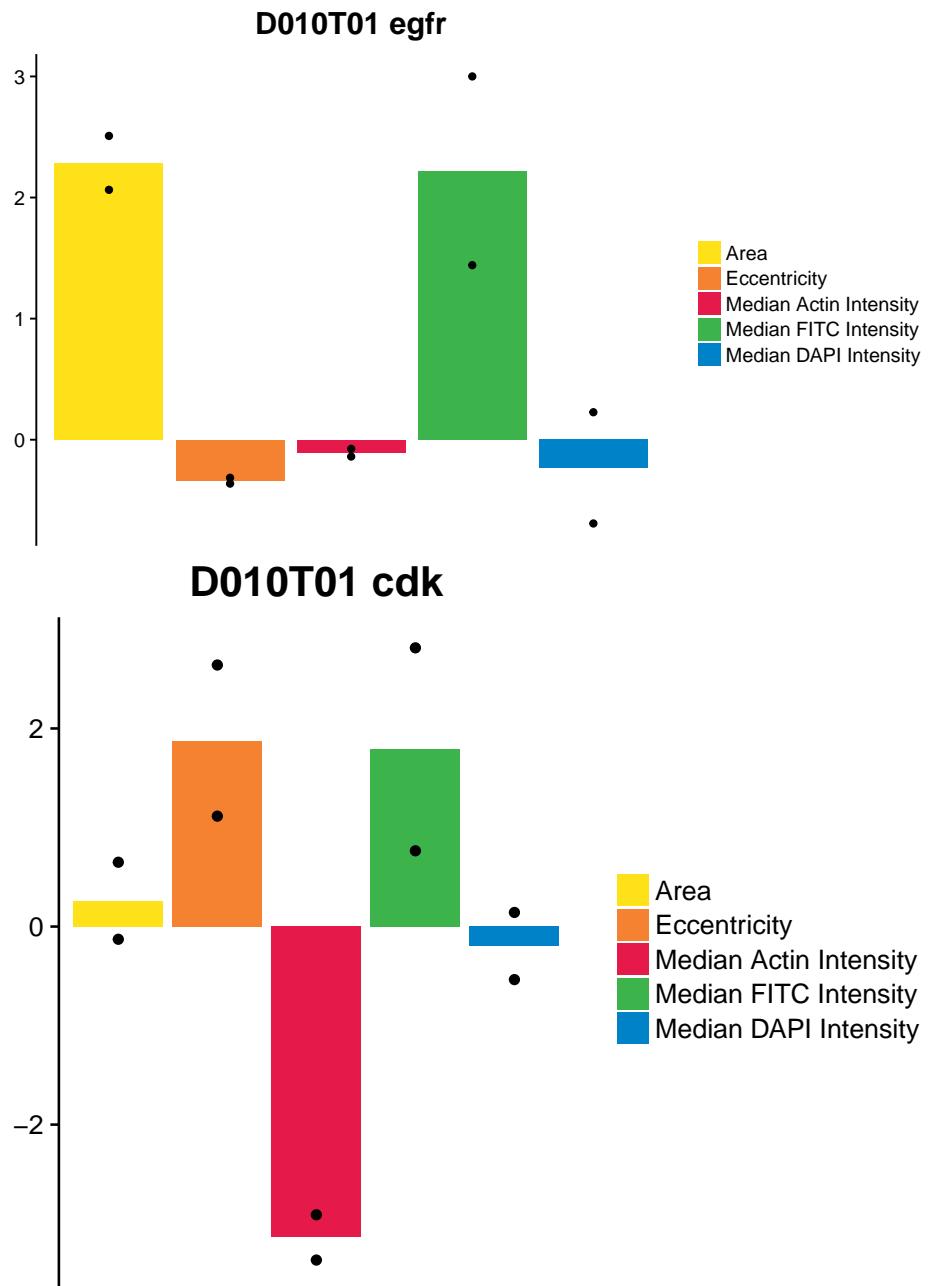
Drug-Induced Phenotypes



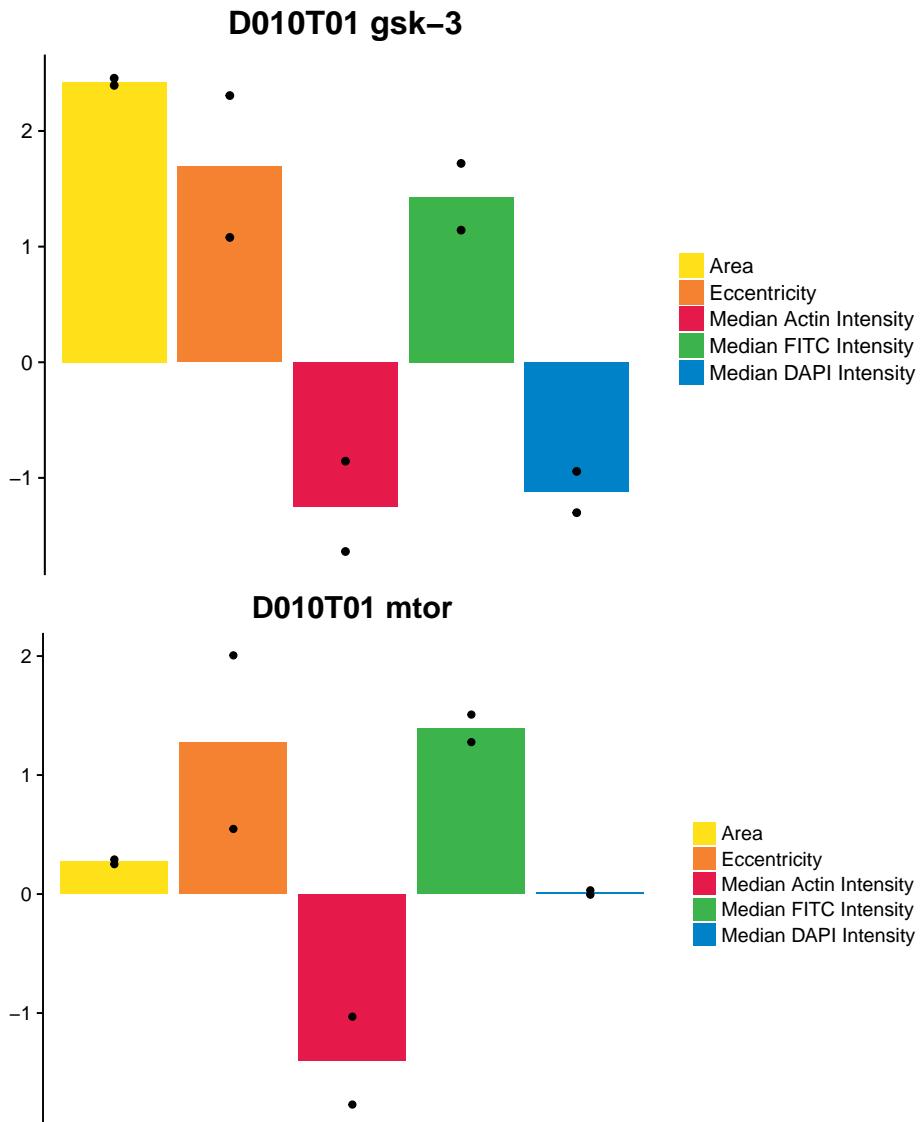
Drug-Induced Phenotypes



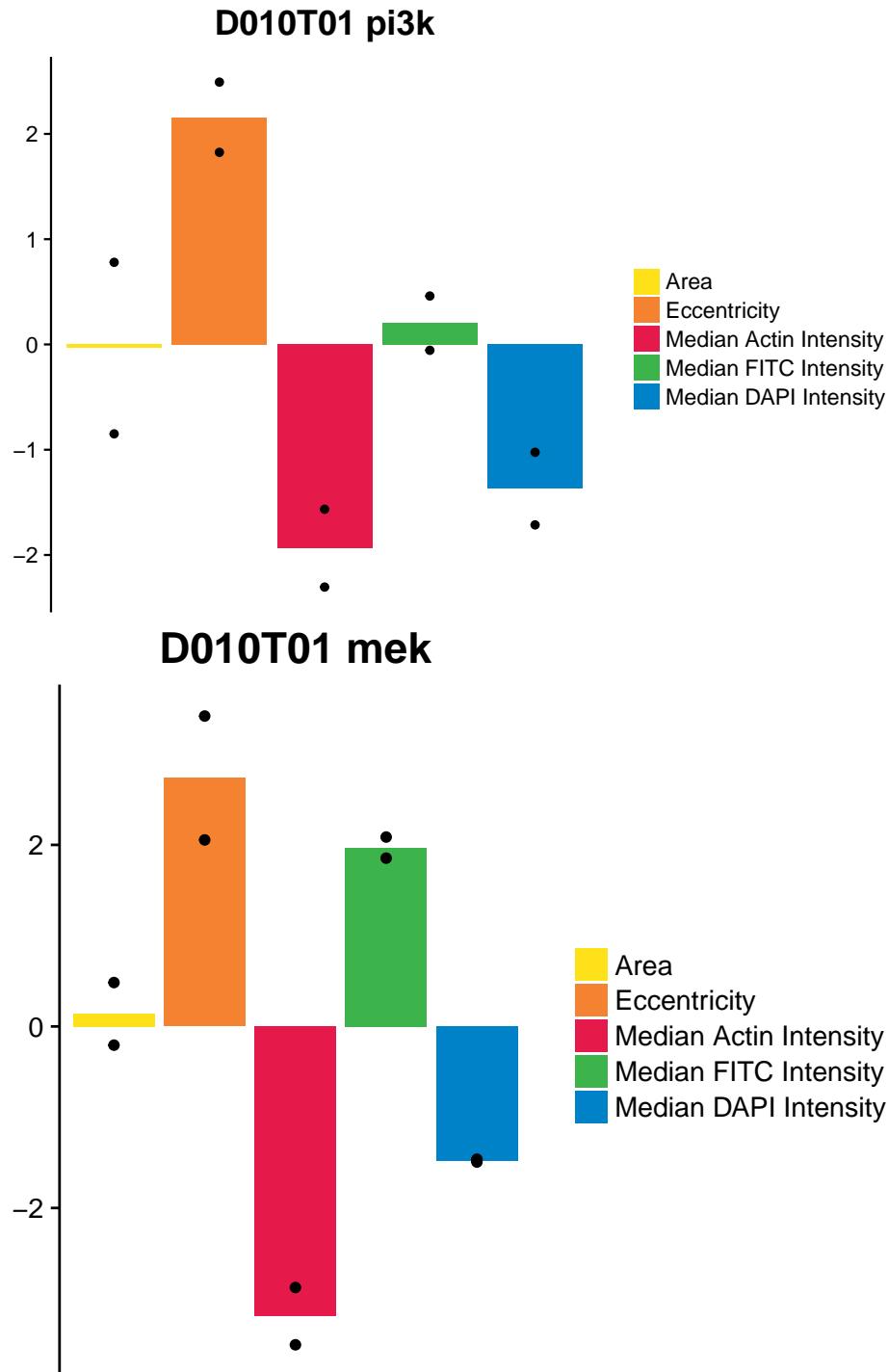
Drug-Induced Phenotypes



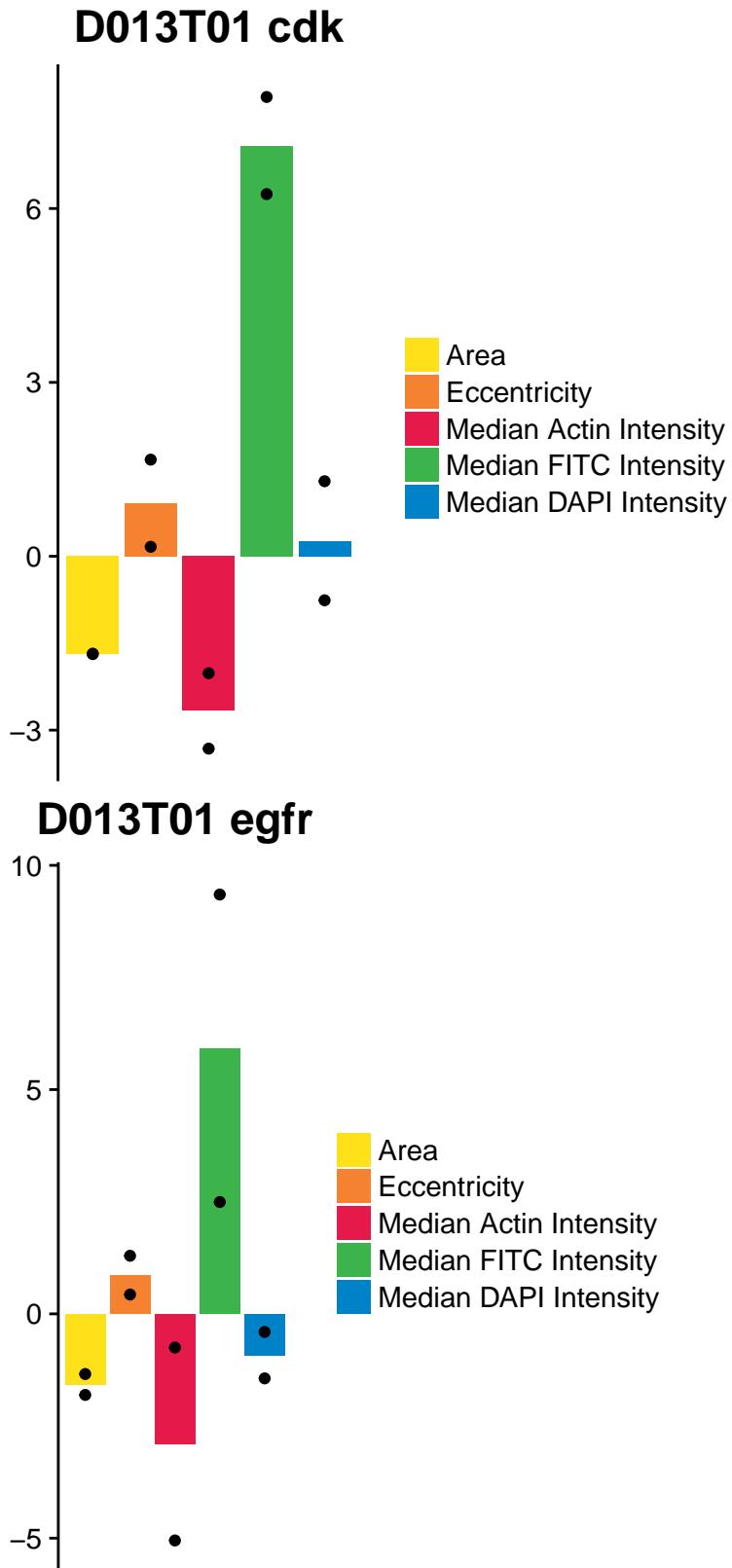
Drug-Induced Phenotypes



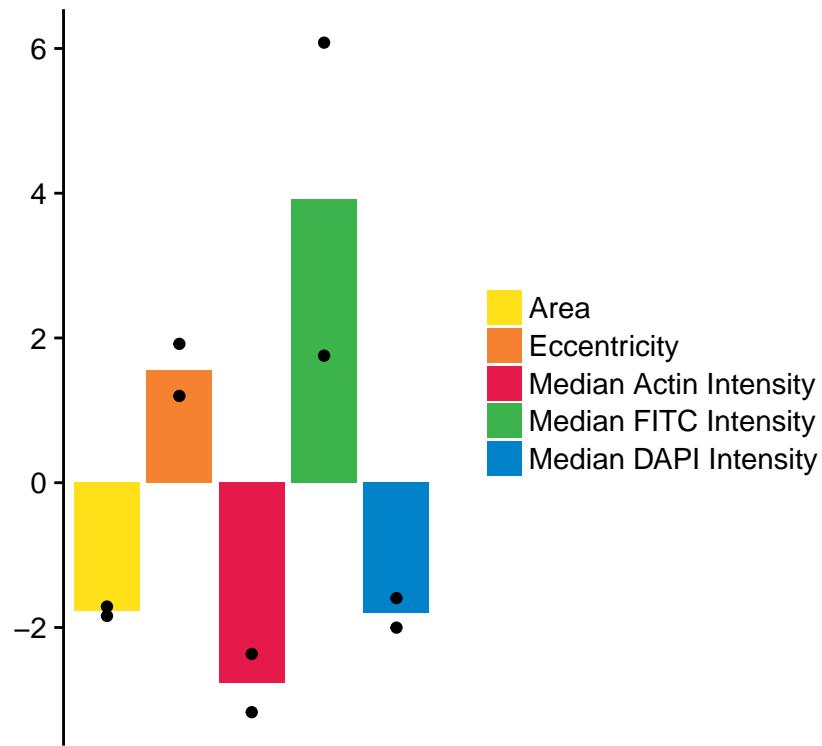
Drug-Induced Phenotypes



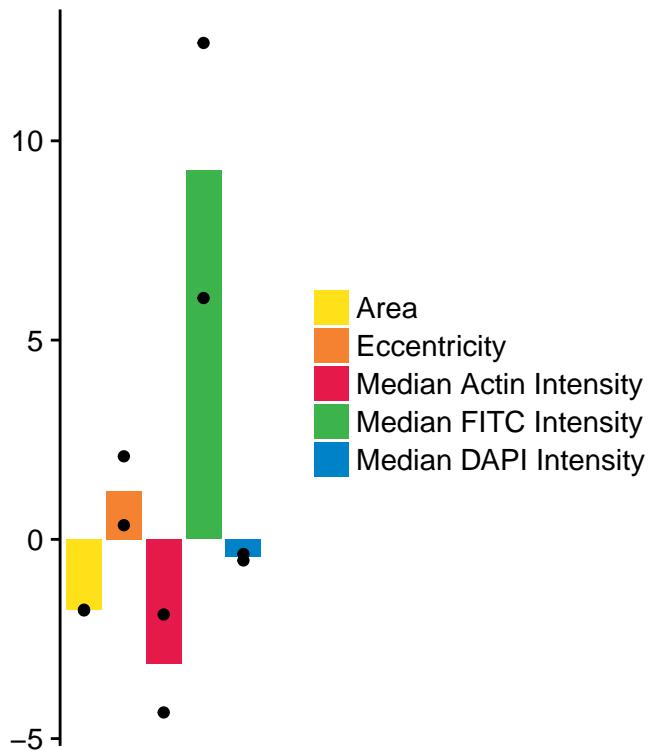
Drug-Induced Phenotypes



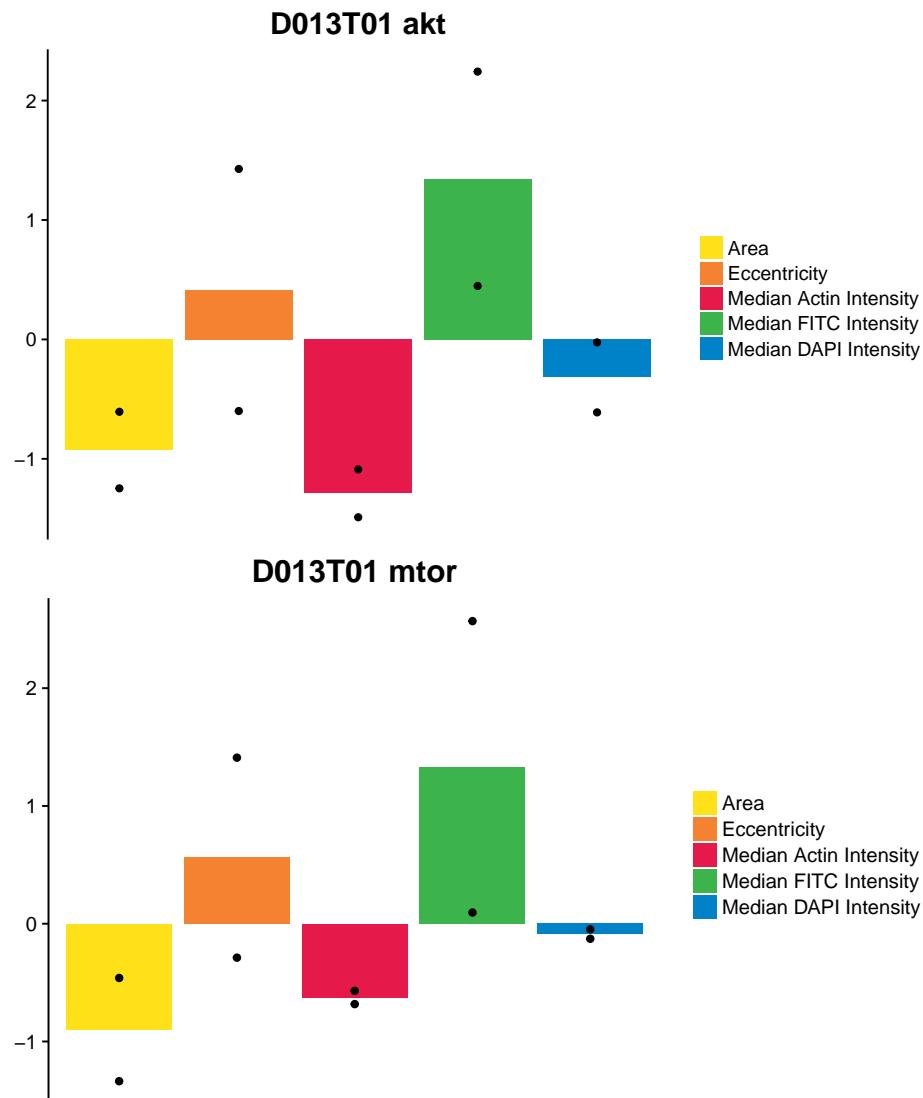
D013T01 aurora kinase



D013T01 atm

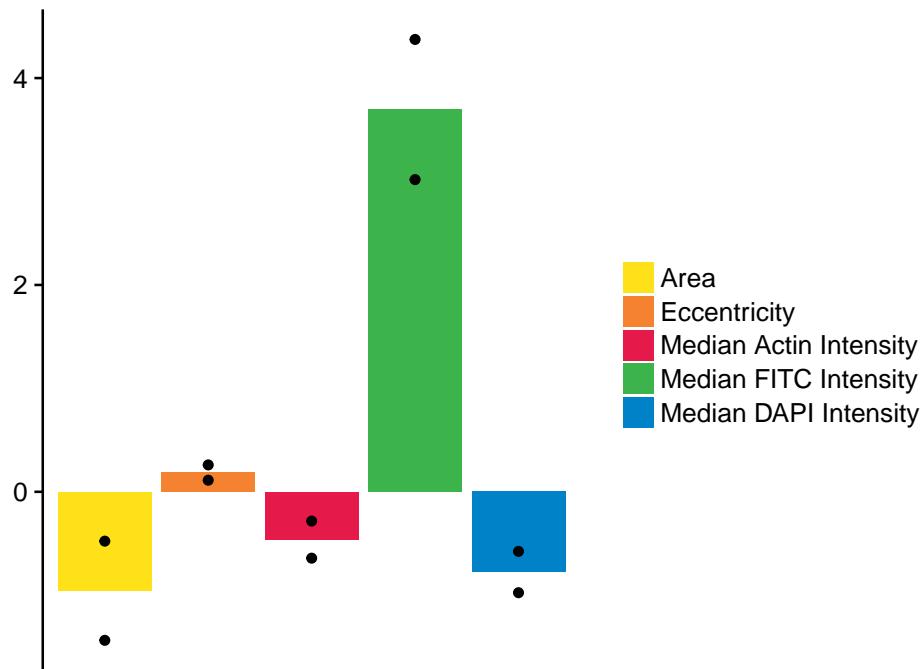


Drug-Induced Phenotypes

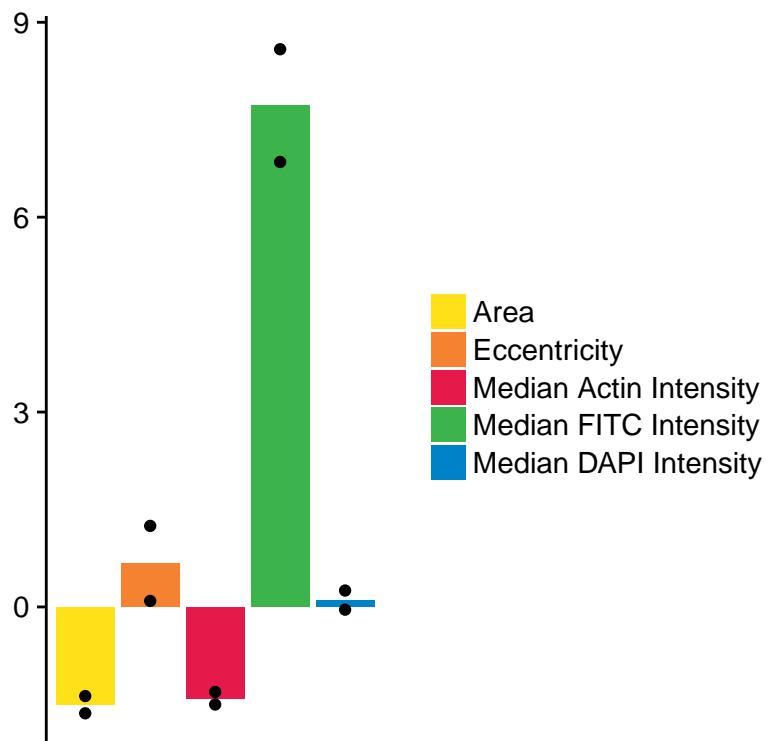


Drug-Induced Phenotypes

D013T01 mek

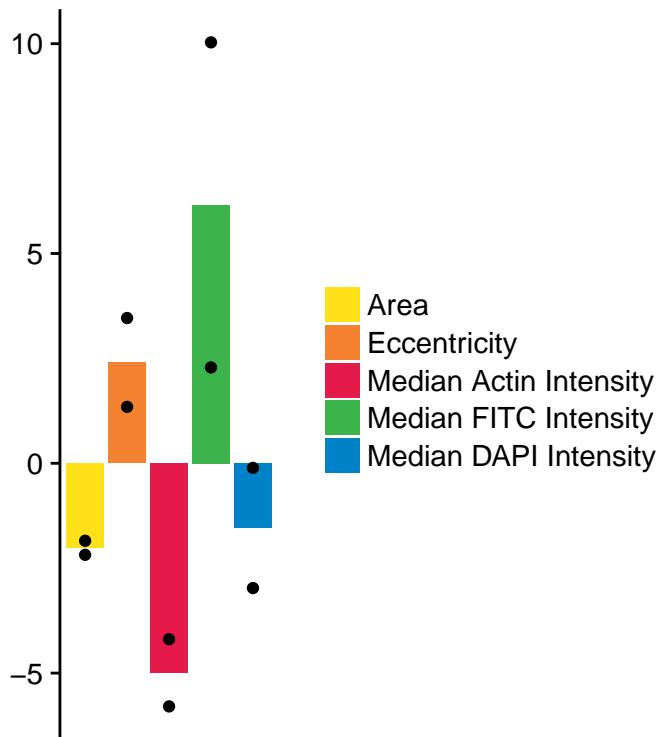


D013T01 pi3k

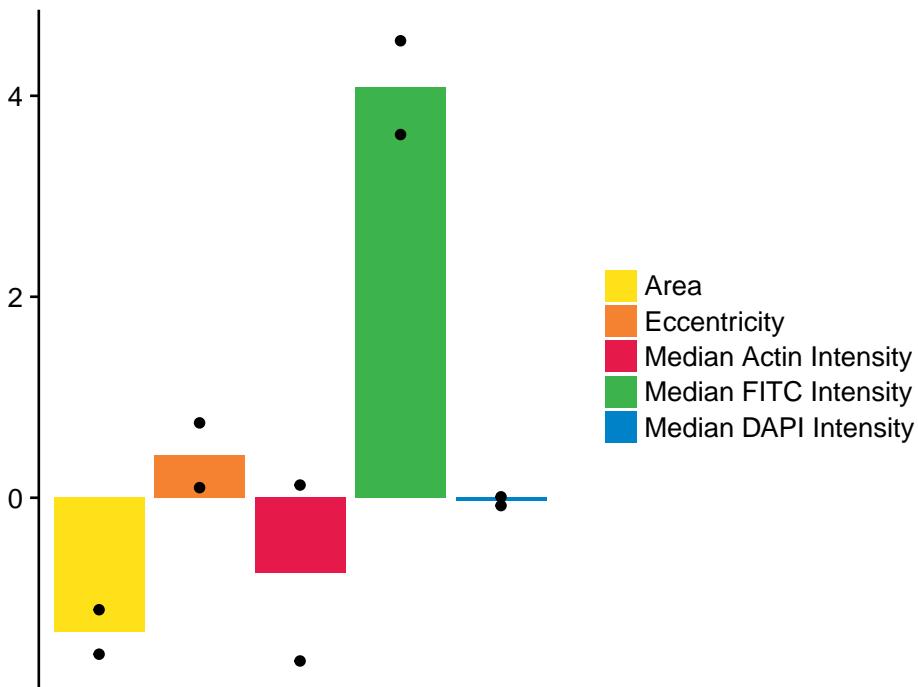


Drug-Induced Phenotypes

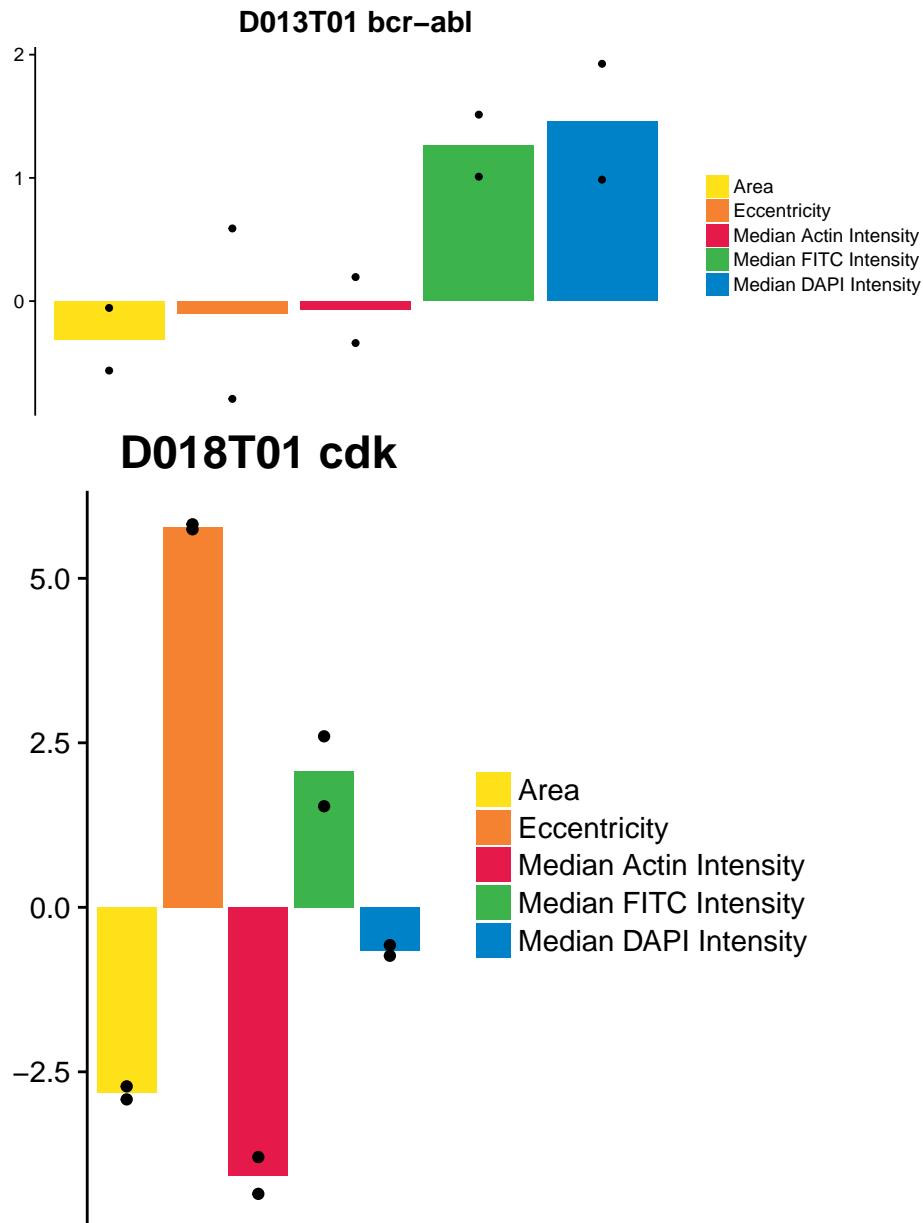
D013T01 plk



D013T01 igf-1r

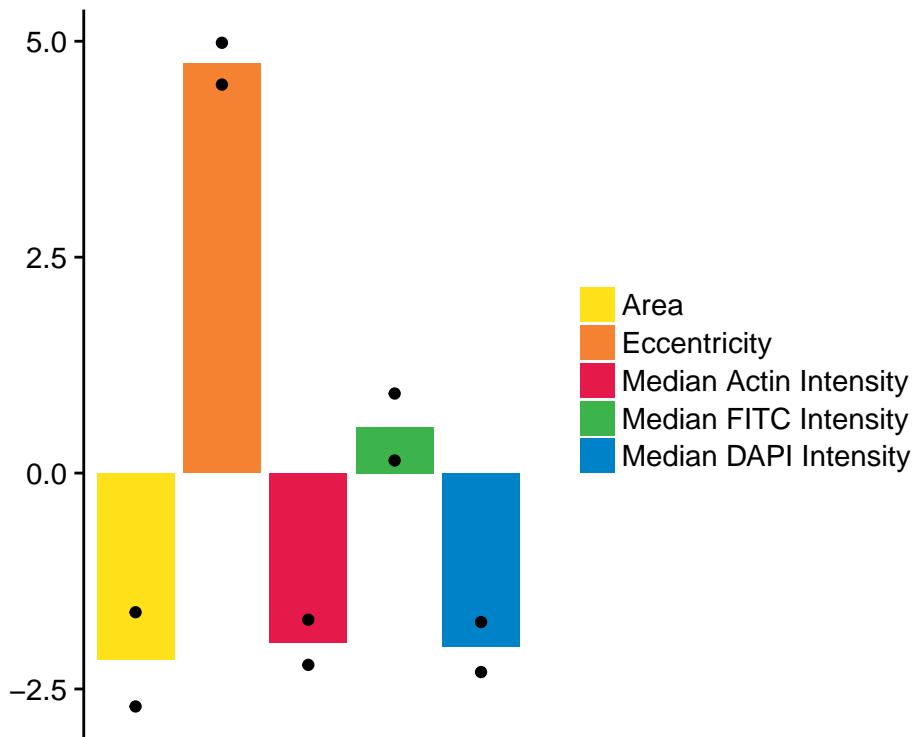


Drug-Induced Phenotypes

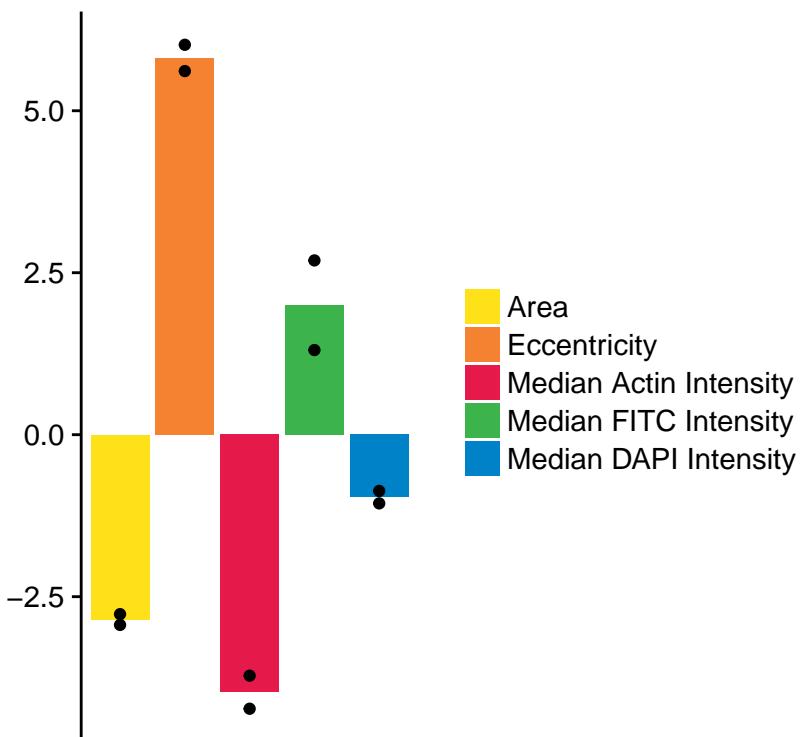


Drug-Induced Phenotypes

D018T01 egfr

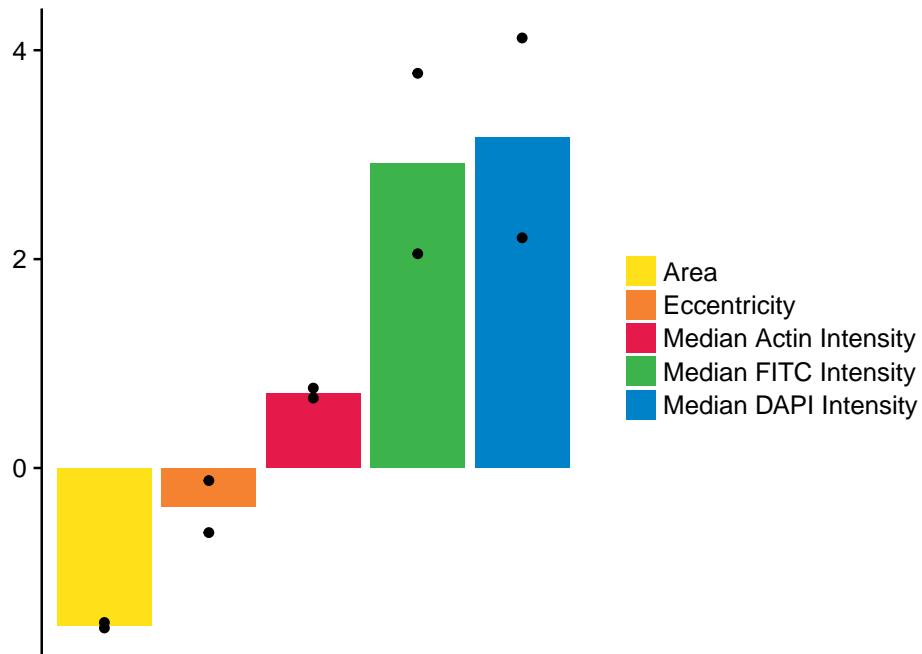


D018T01 jak

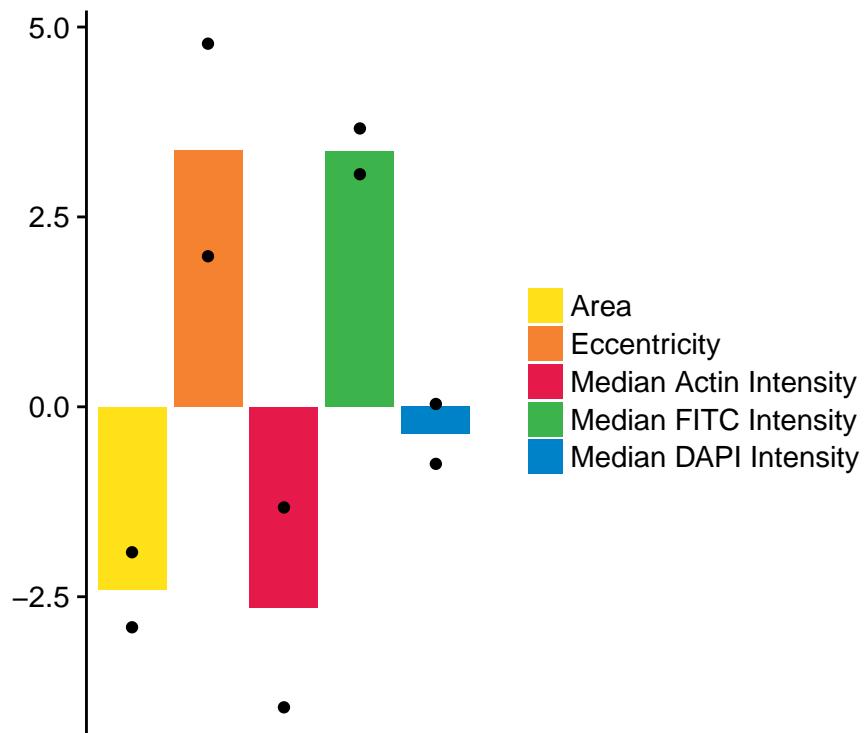


Drug-Induced Phenotypes

D018T01 mtor

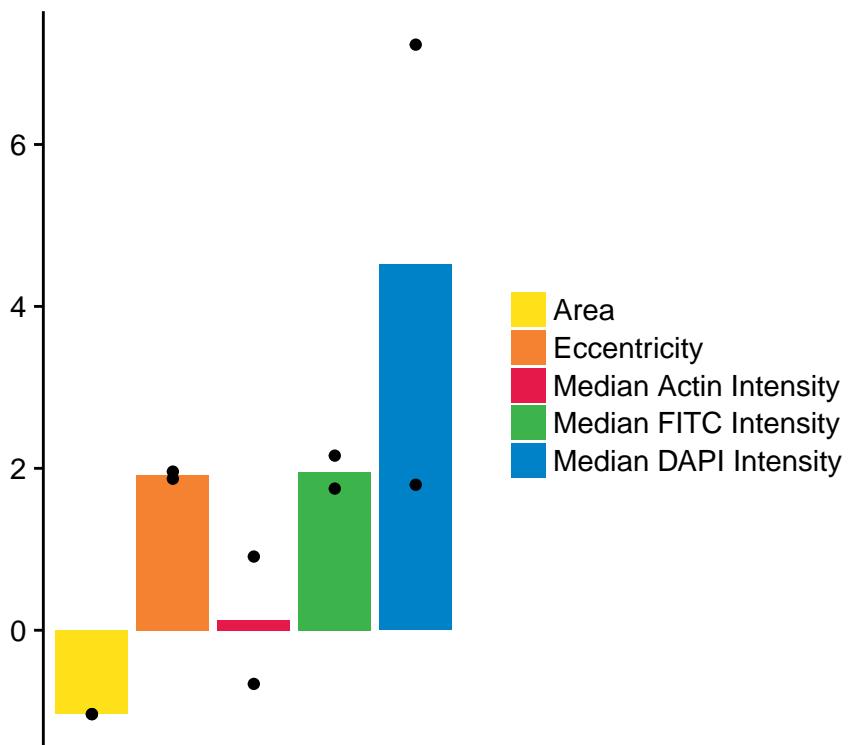


D018T01 gsk-3

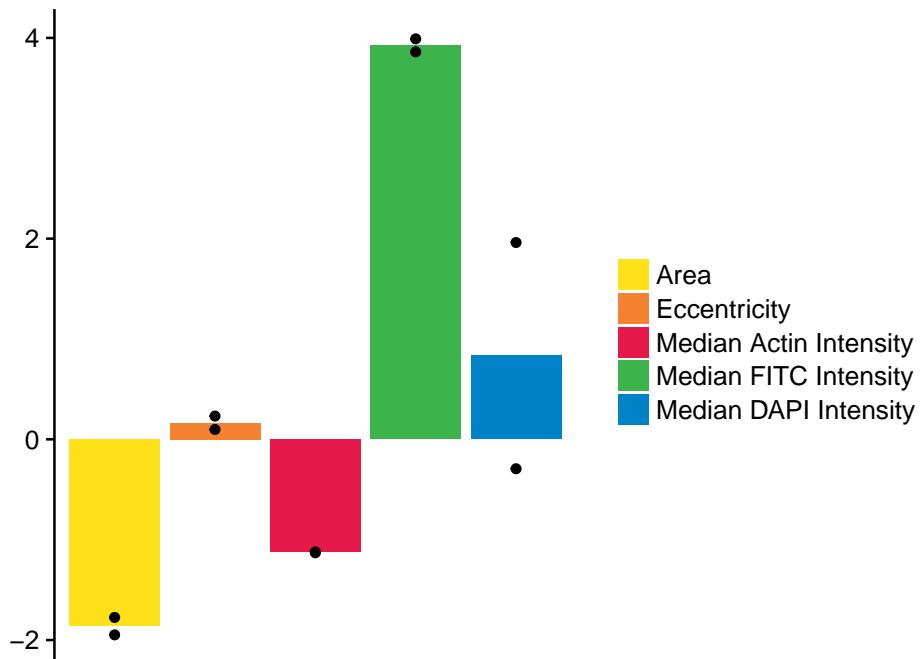


Drug-Induced Phenotypes

D018T01 akt

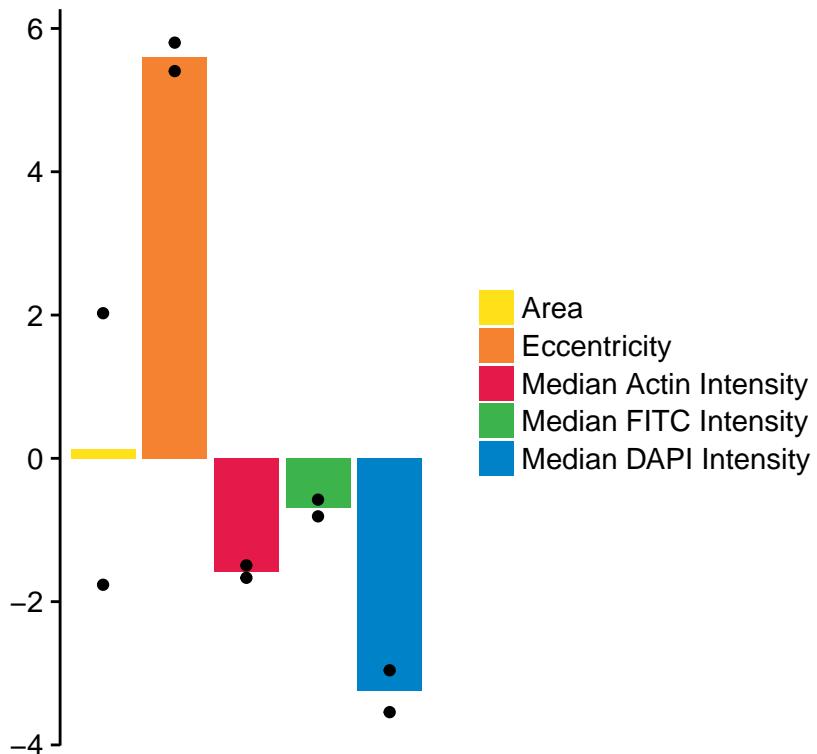


D018T01 pi3k

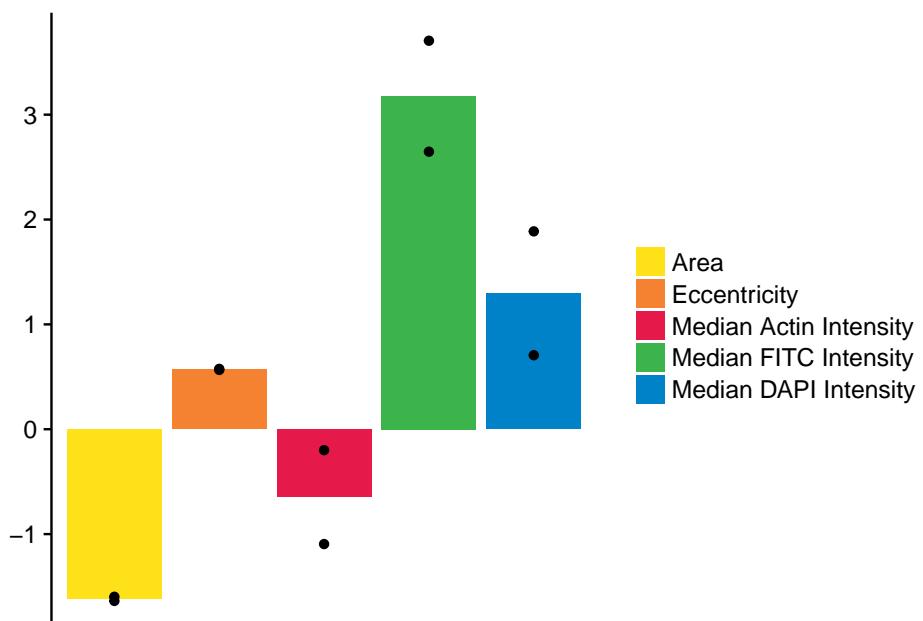


Drug-Induced Phenotypes

D018T01 mek

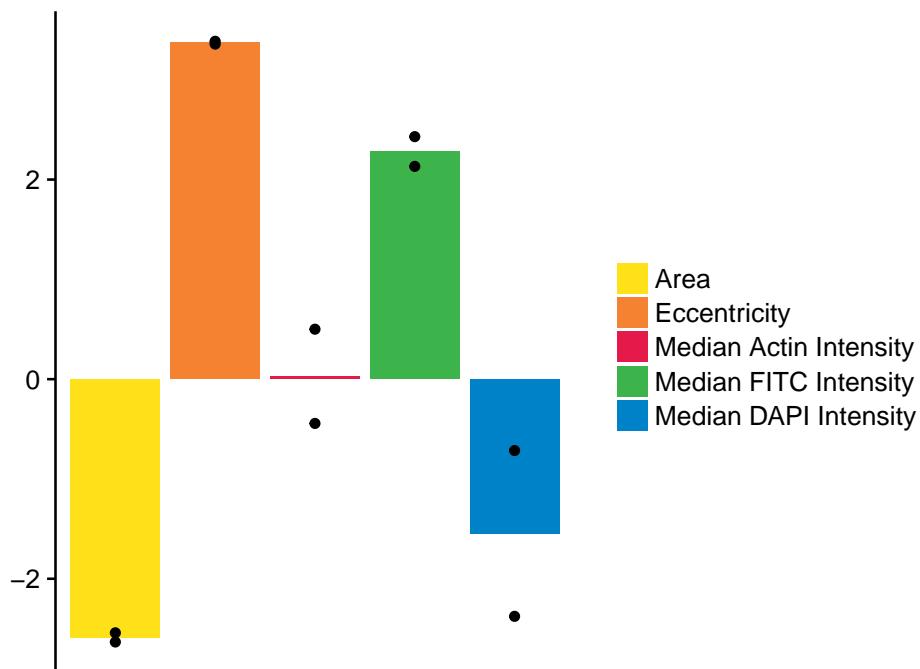


D018T01 pdgfr

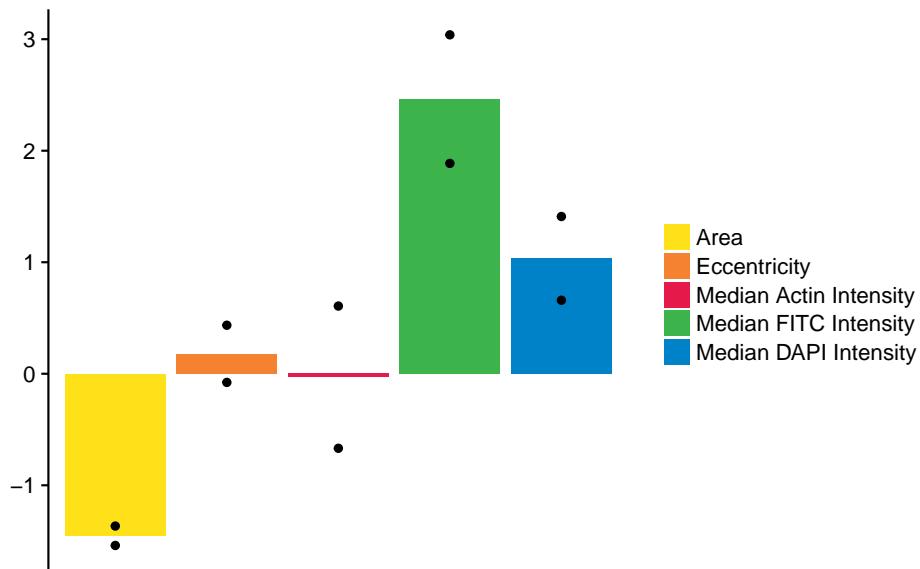


Drug-Induced Phenotypes

D018T01 plk

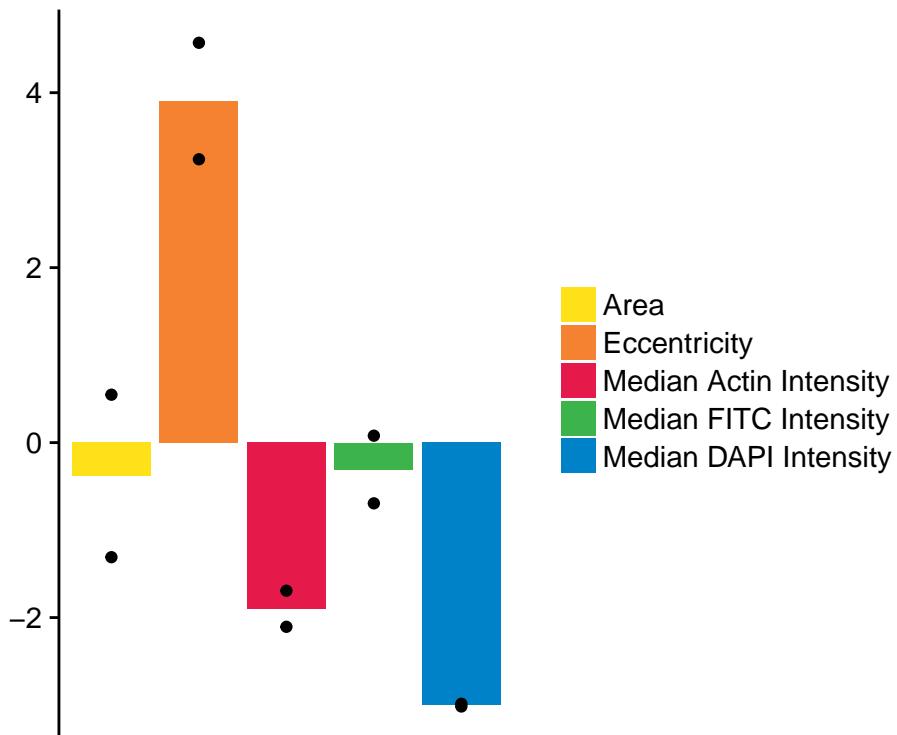


D018T01 bcr-abl

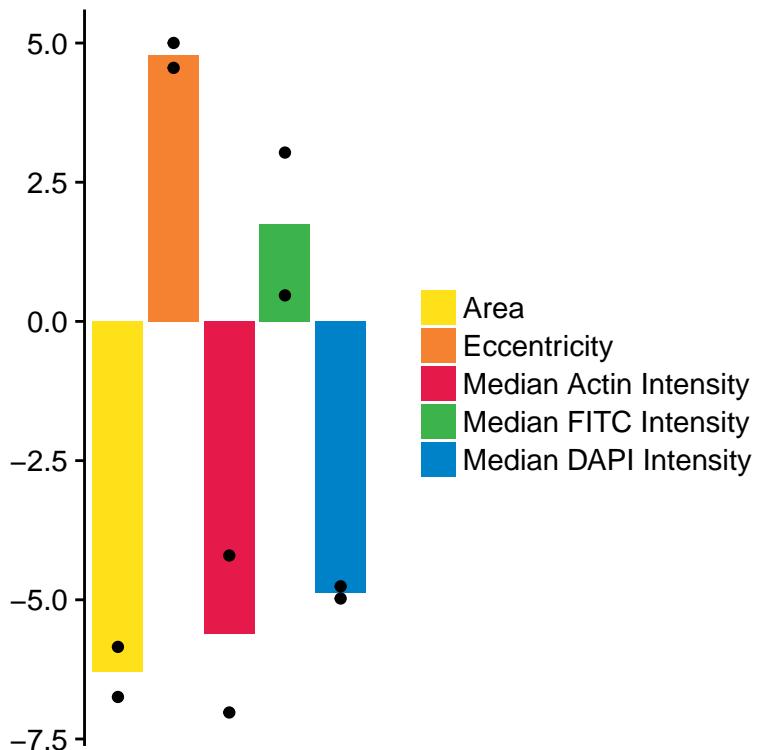


Drug-Induced Phenotypes

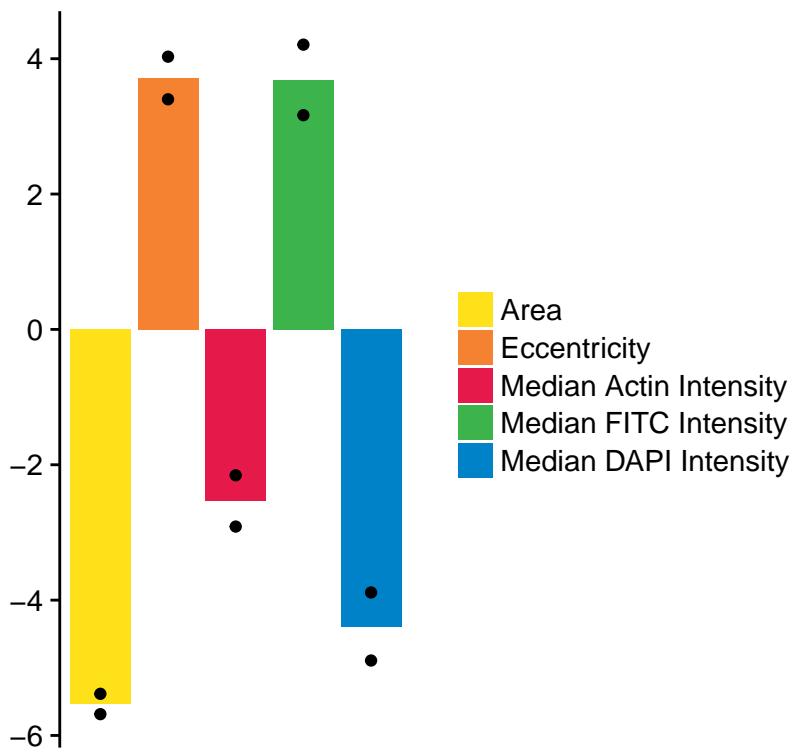
D018T01 vegfr



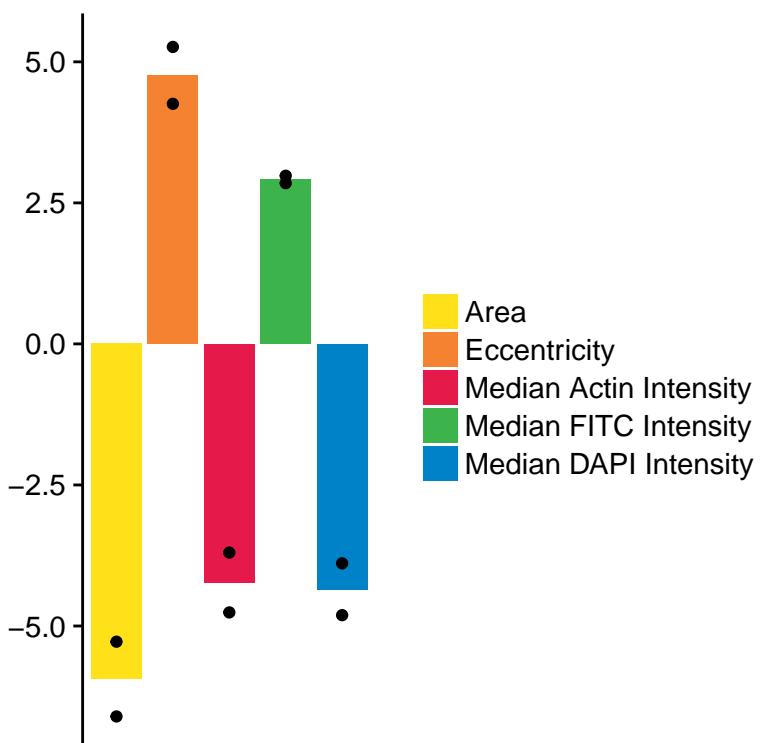
D019T01 aurora kinase



D019T01 alk

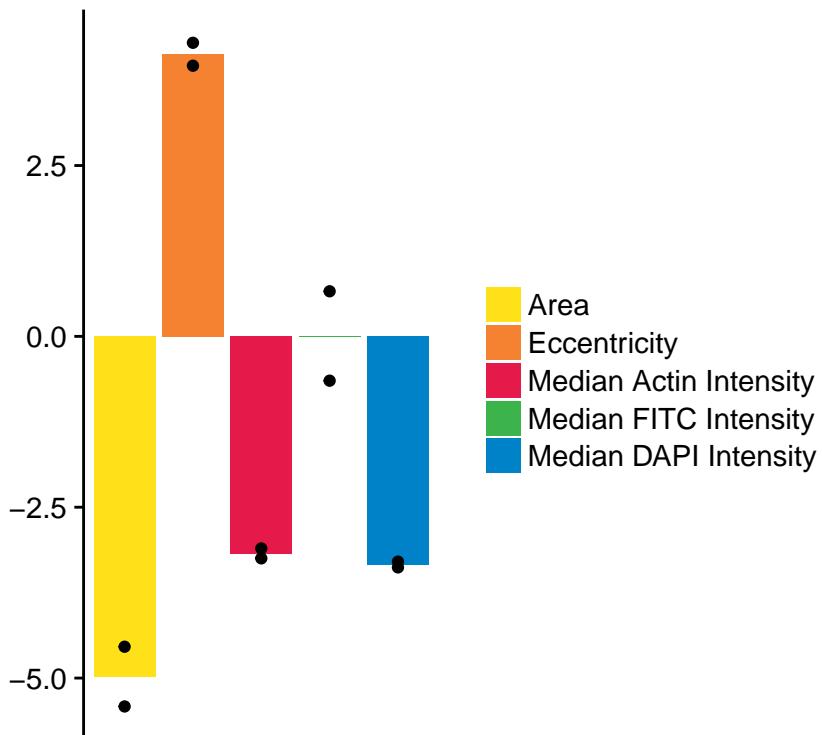


D019T01 atm

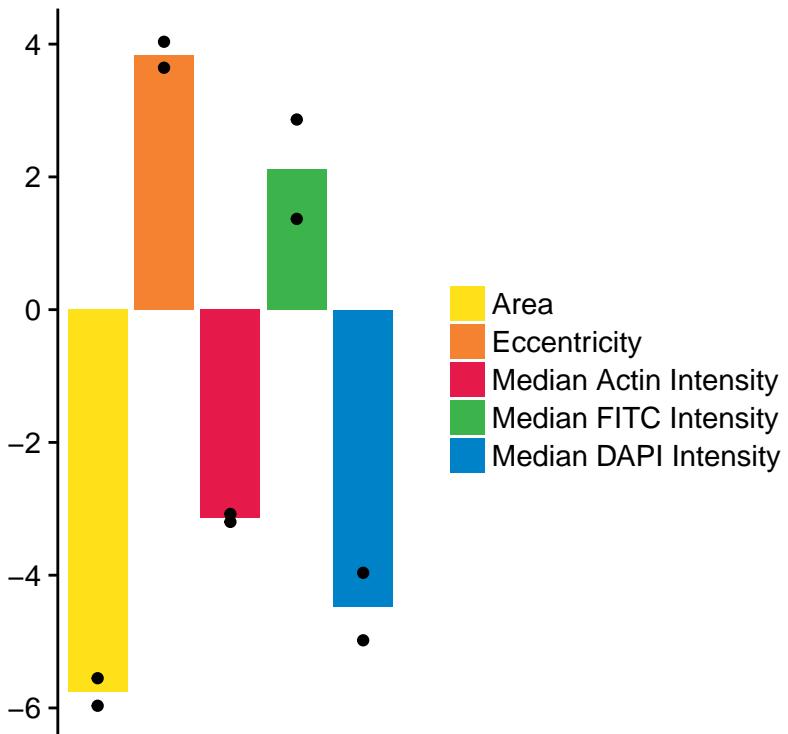


Drug-Induced Phenotypes

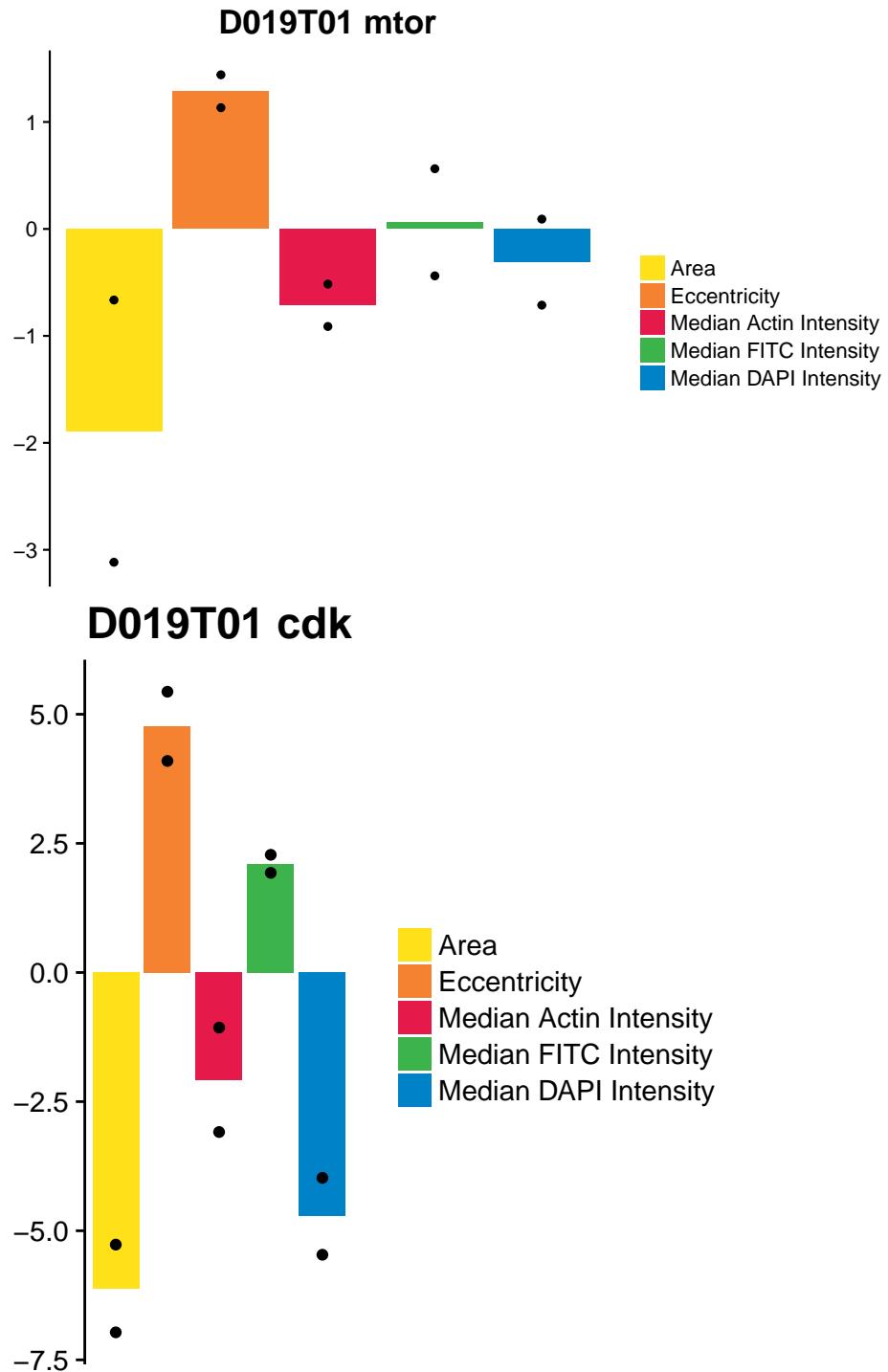
D019T01 gsk-3



D019T01 jak

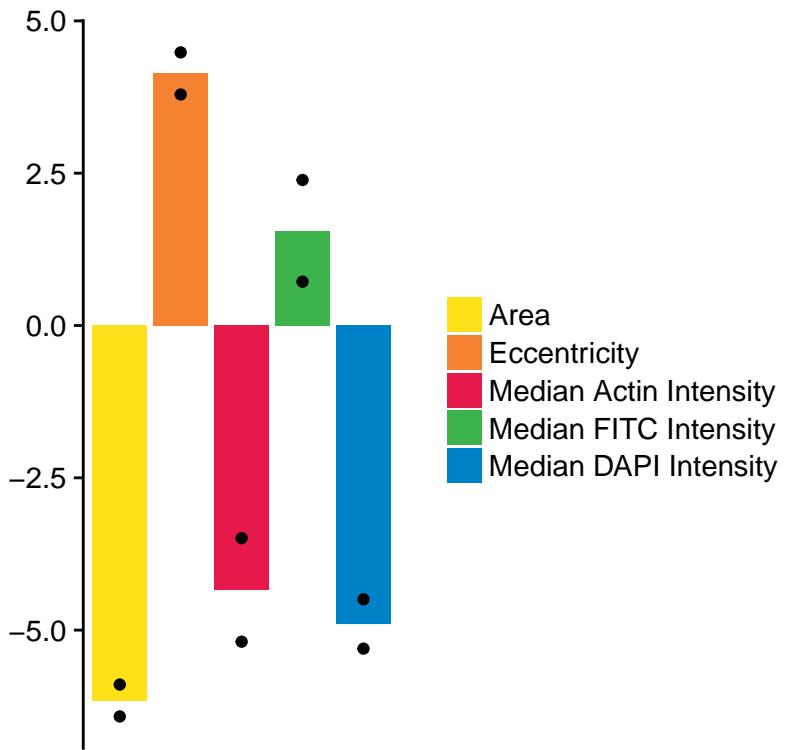


Drug-Induced Phenotypes

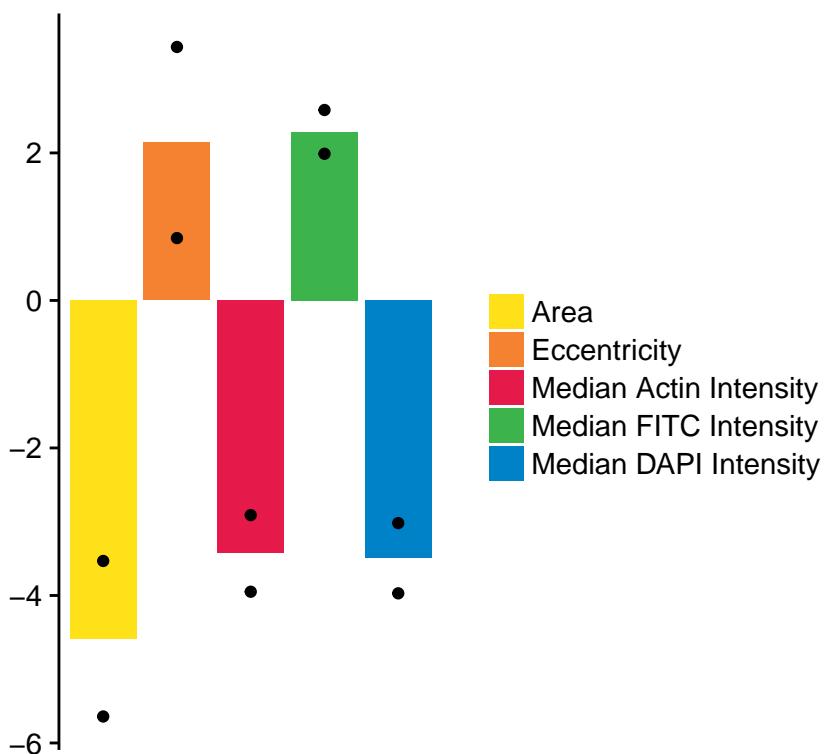


Drug-Induced Phenotypes

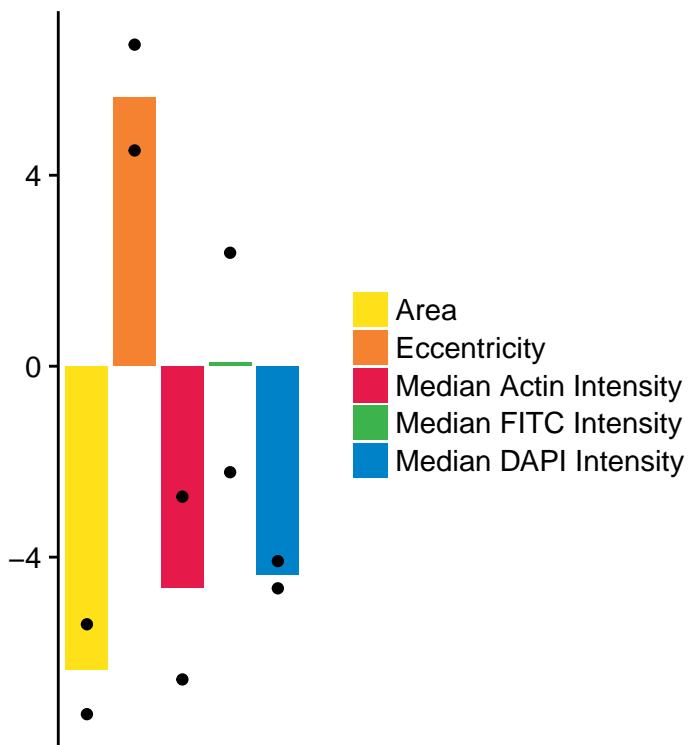
D019T01 plk



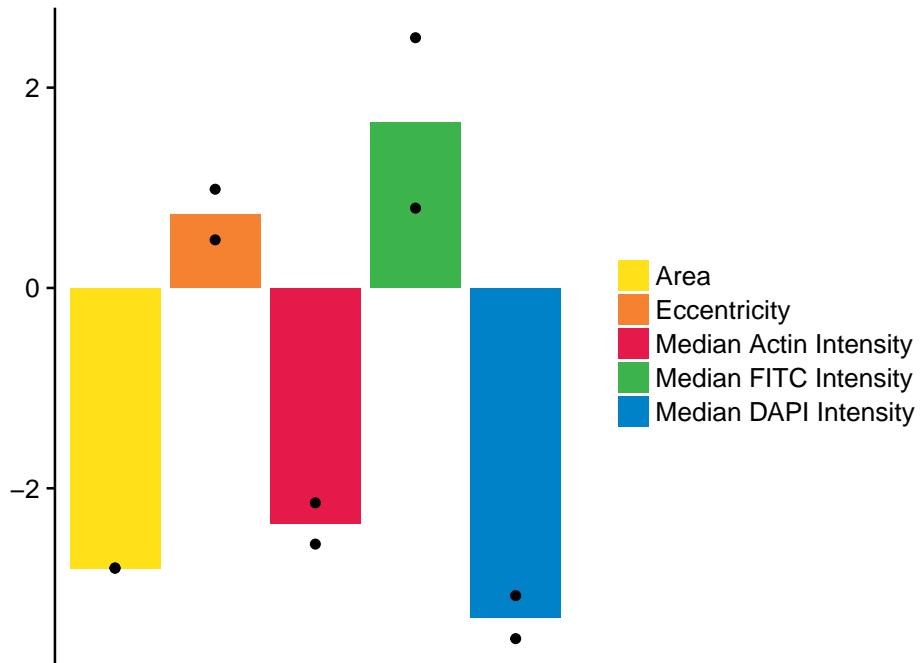
D019T01 mek



D019T01 c-met

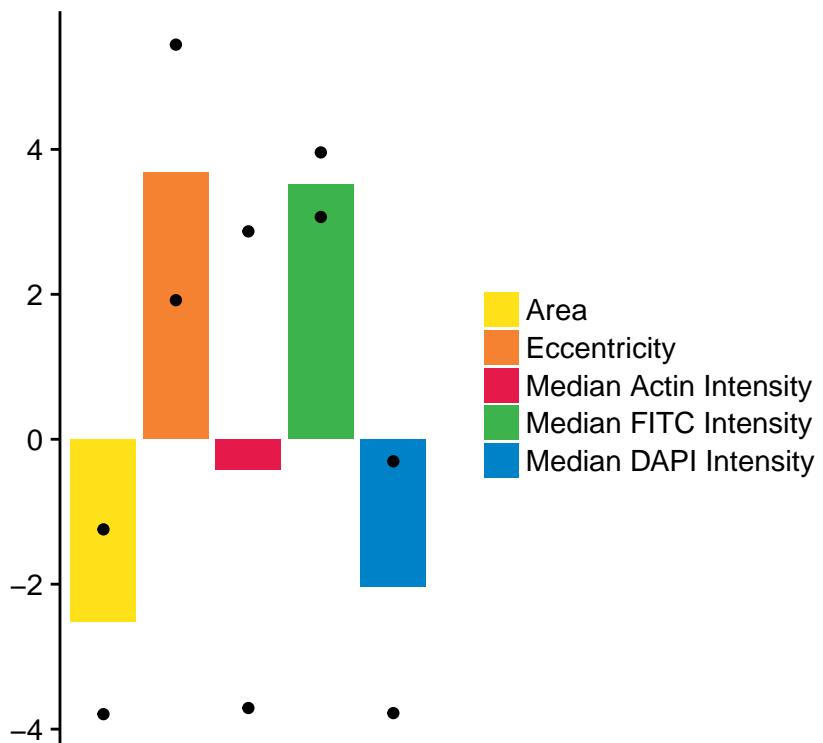


D019T01 erk

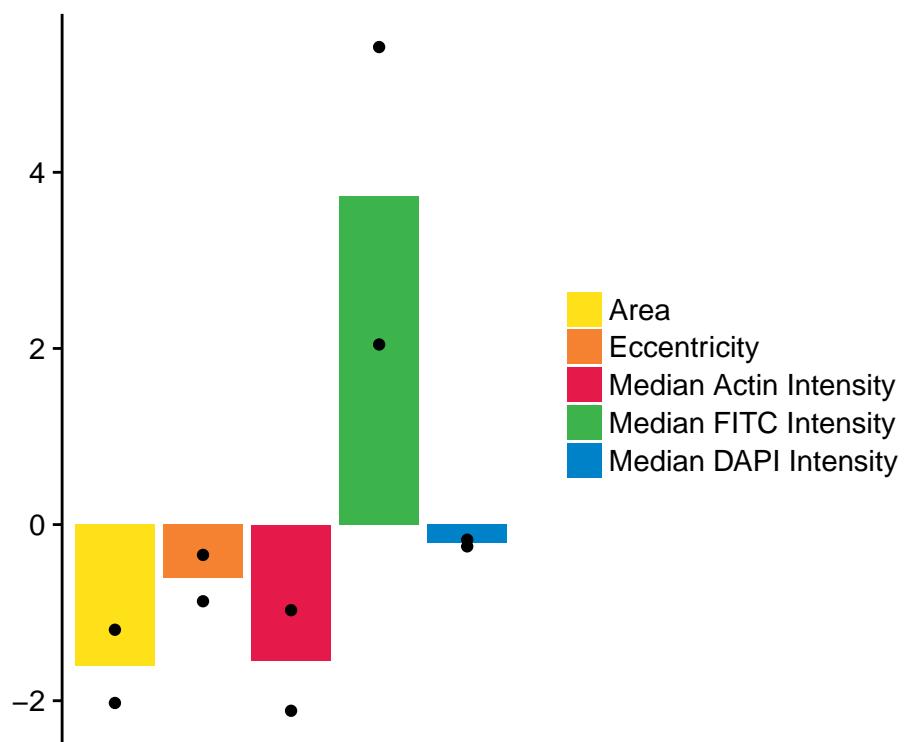


Drug-Induced Phenotypes

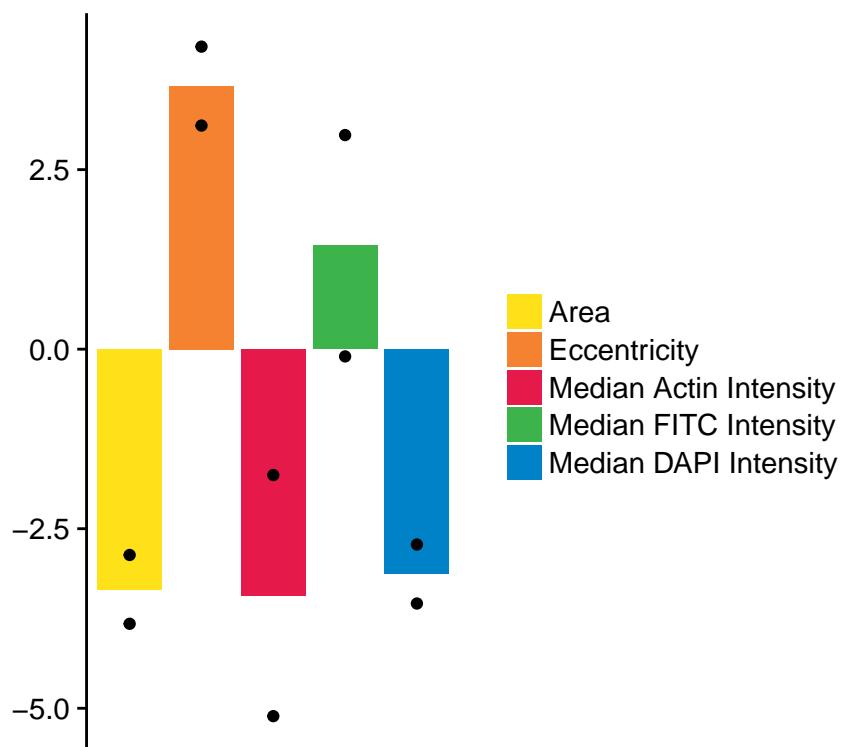
D020T01 cdk



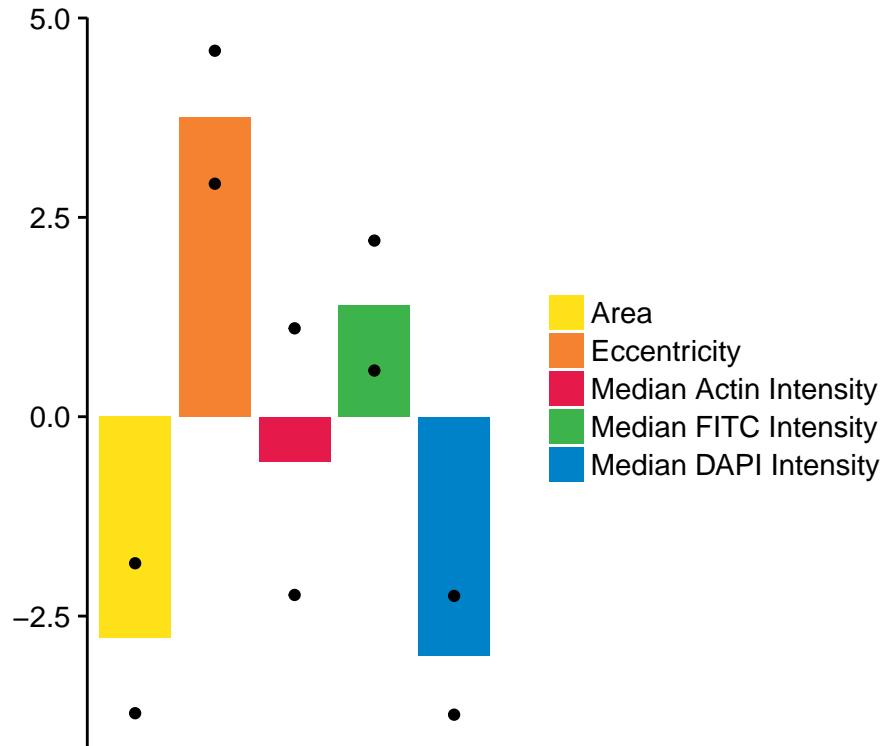
D020T01 egfr



D020T01 aurora kinase

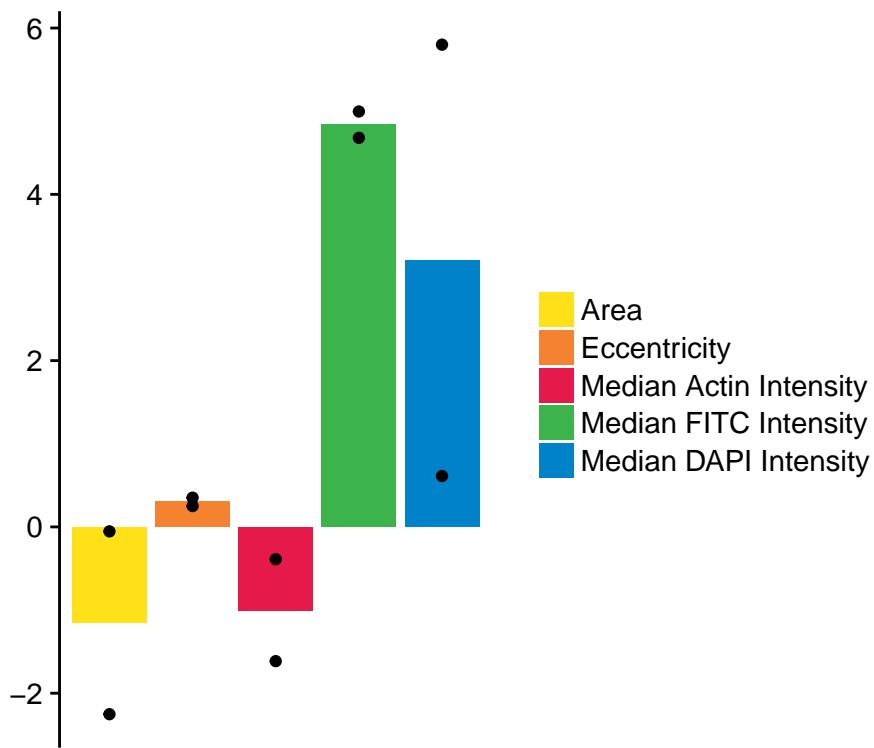


D020T01 jak

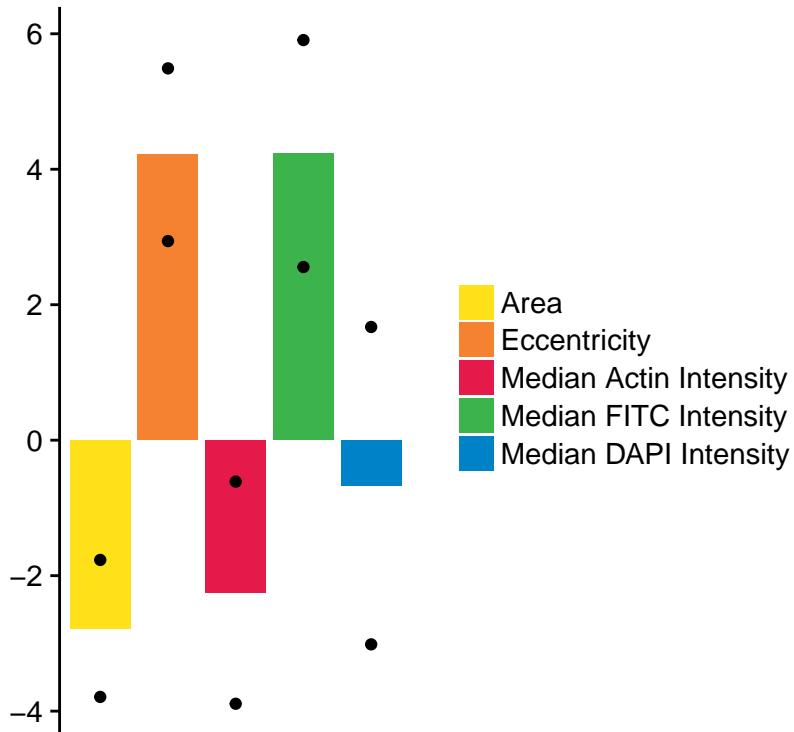


Drug-Induced Phenotypes

D020T01 mtor

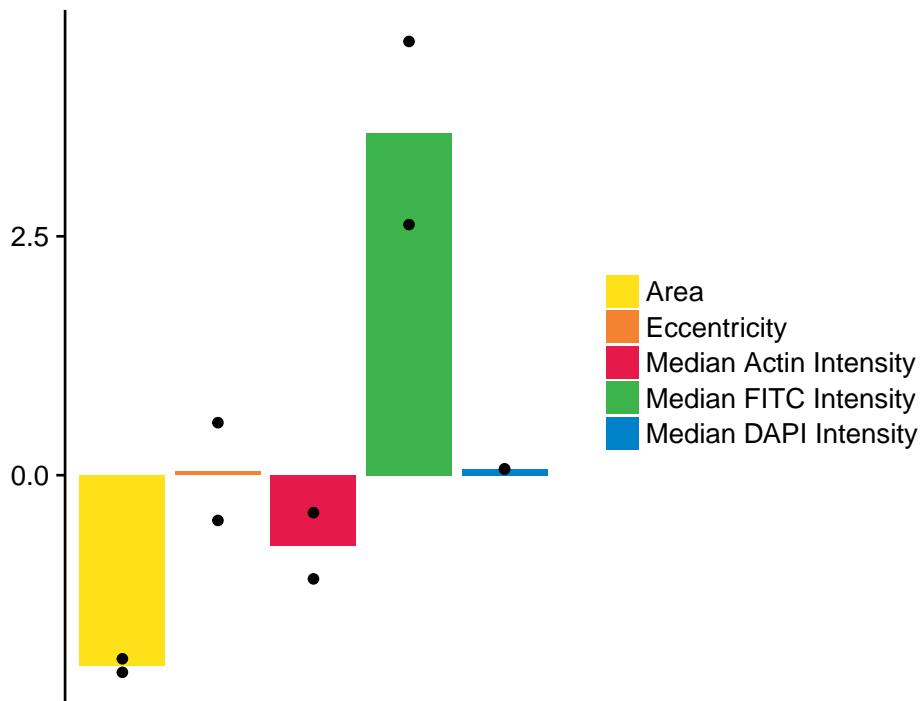


D020T01 gsk-3

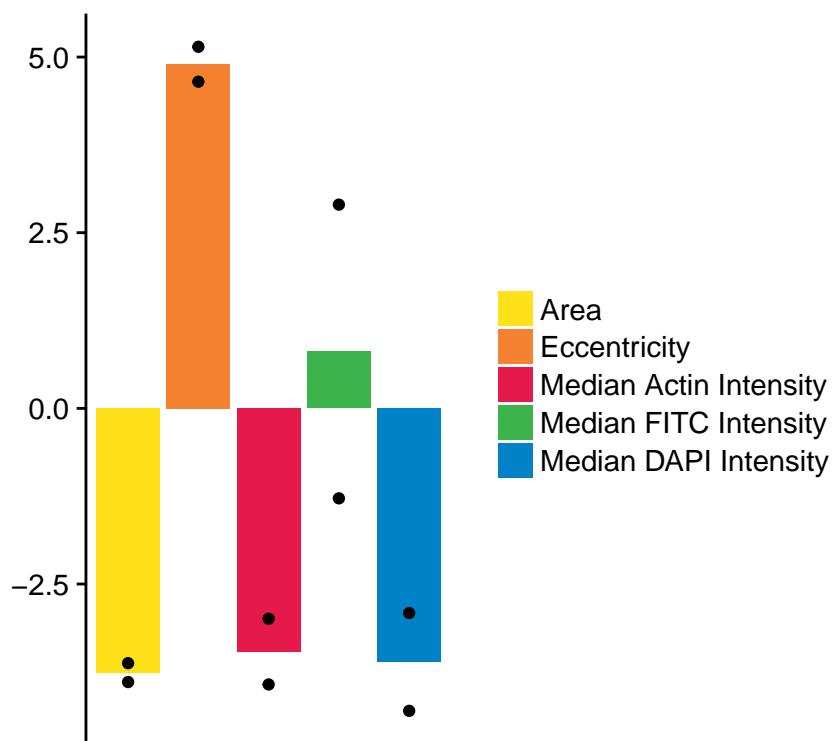


Drug-Induced Phenotypes

D020T01 mek

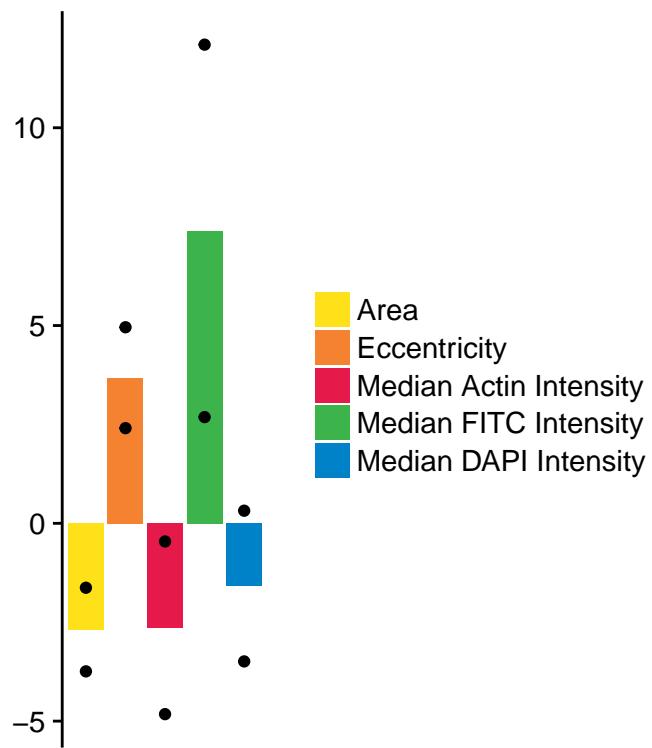


D020T01 plk

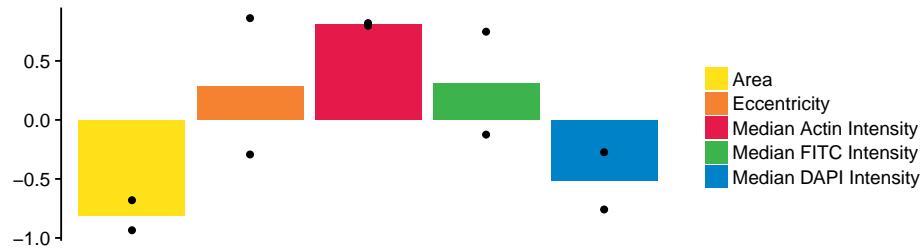


Drug-Induced Phenotypes

D020T01 chk

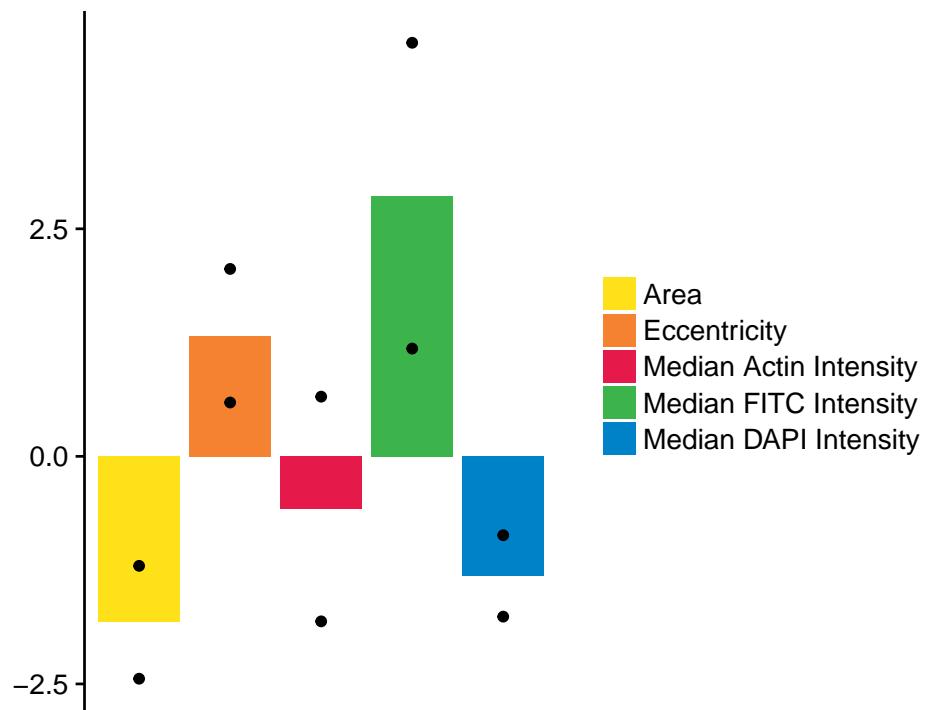


D020T01 bcr-abl

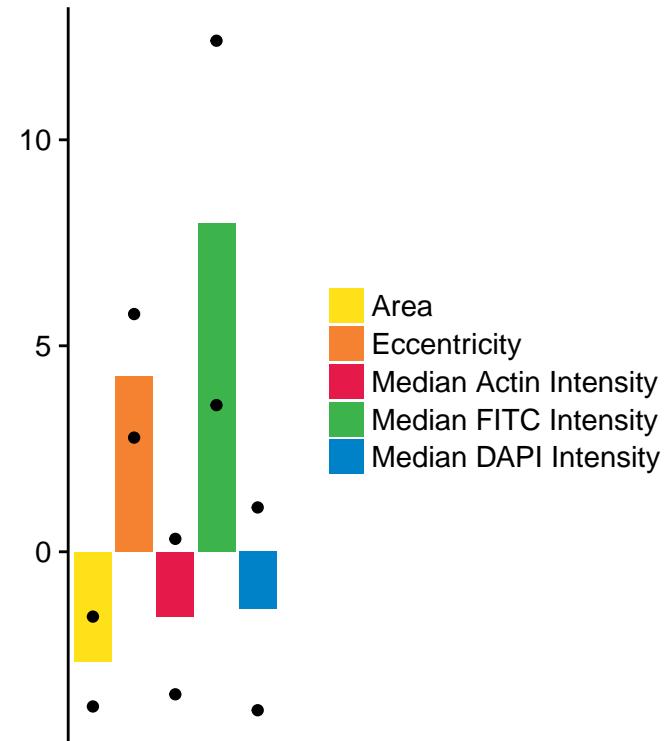


Drug-Induced Phenotypes

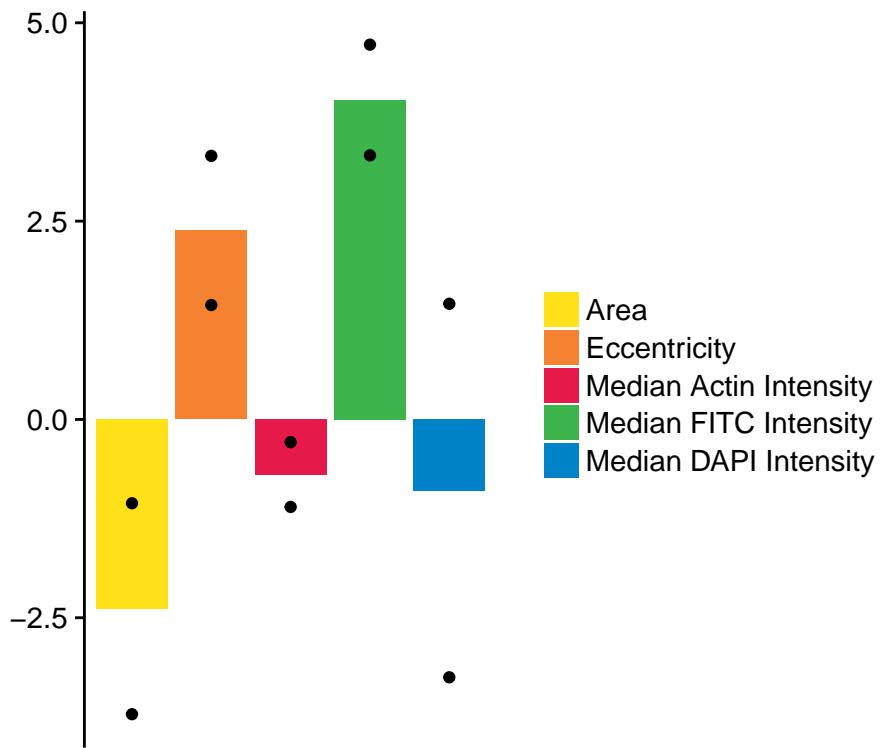
D020T01 vegfr



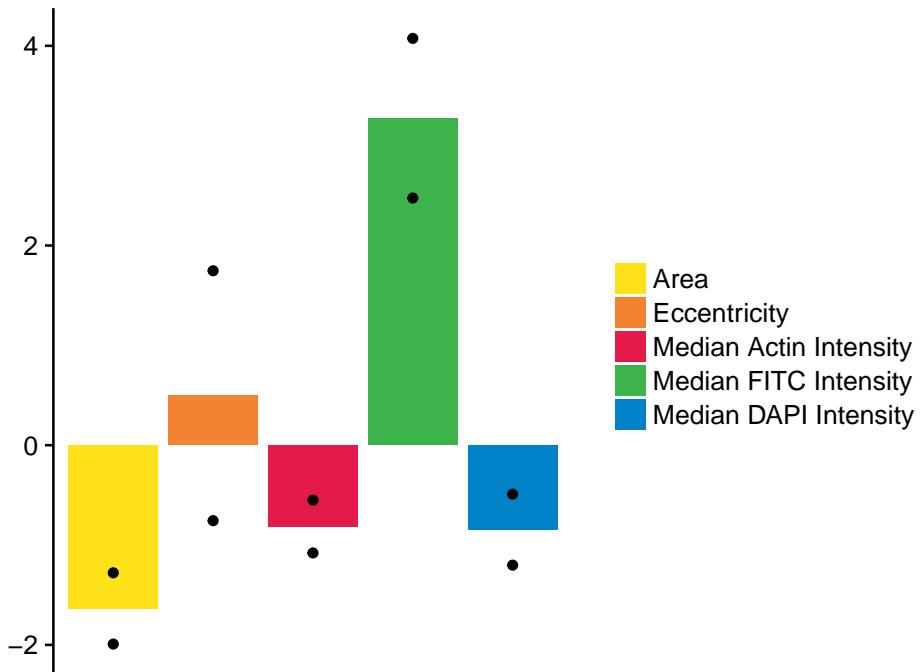
D020T01 c-met



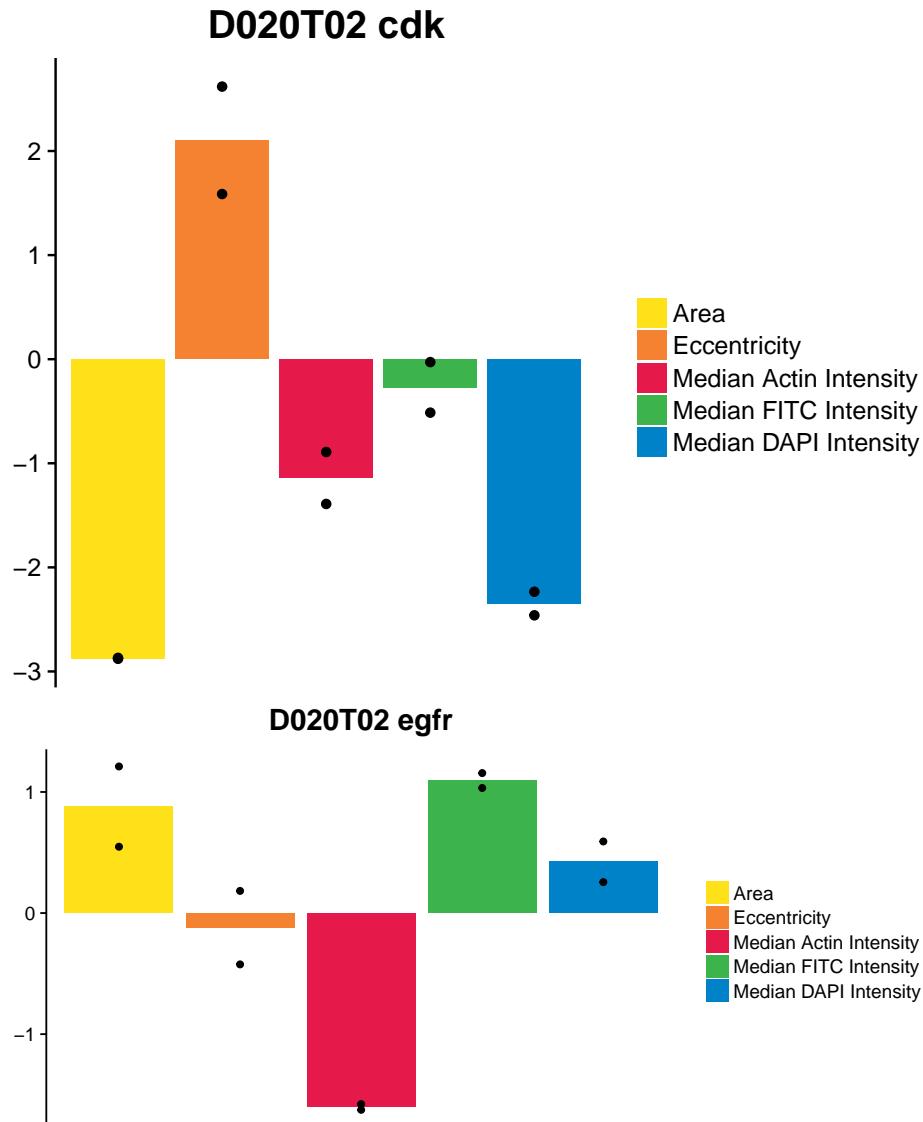
D020T01 fak



D020T01 flt

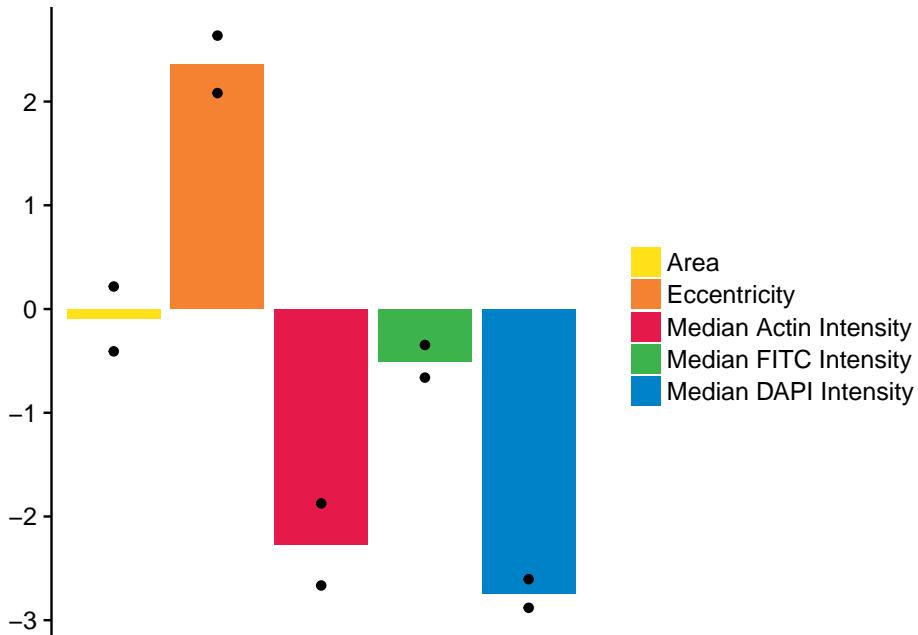


Drug-Induced Phenotypes

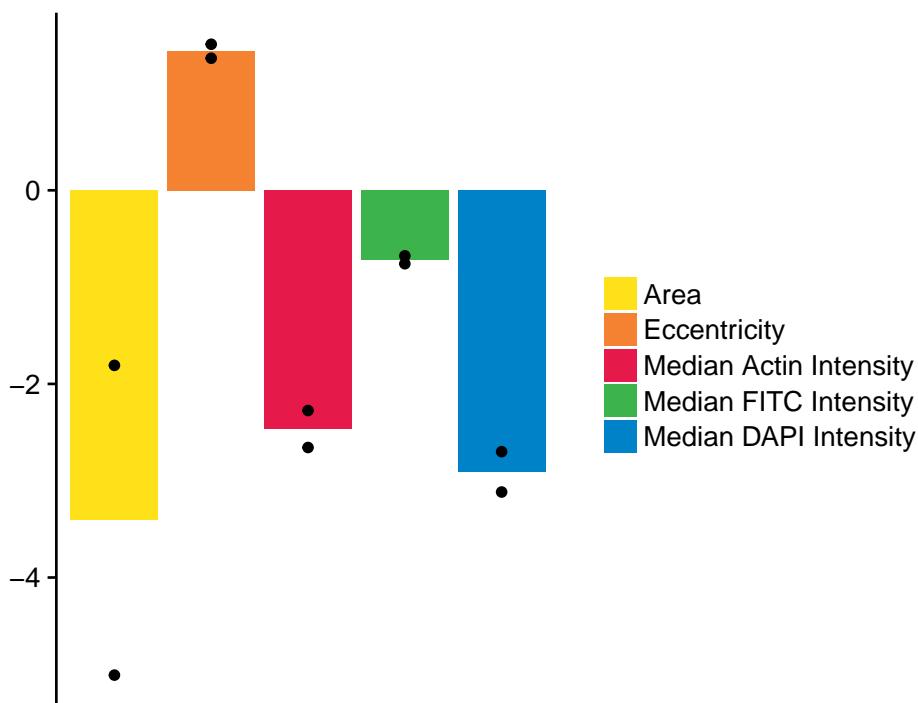


Drug-Induced Phenotypes

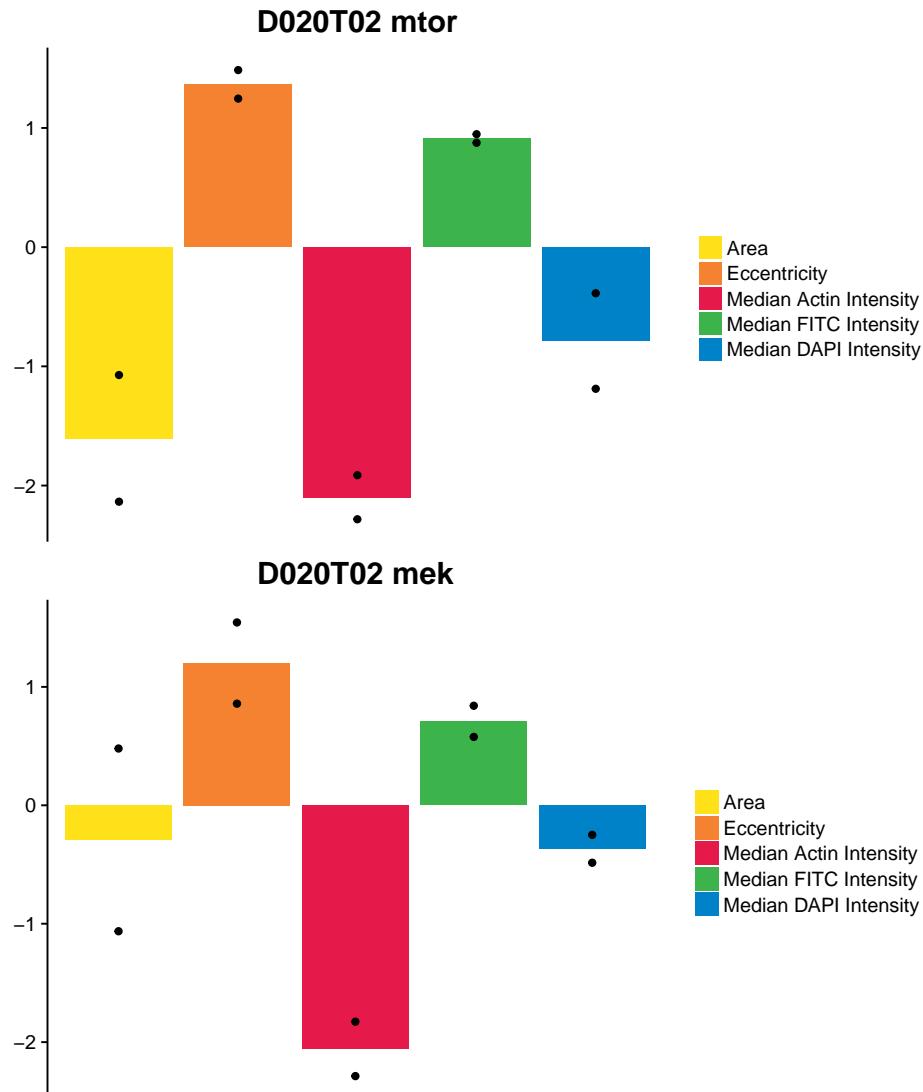
D020T02 aurora kinase



D020T02 alk

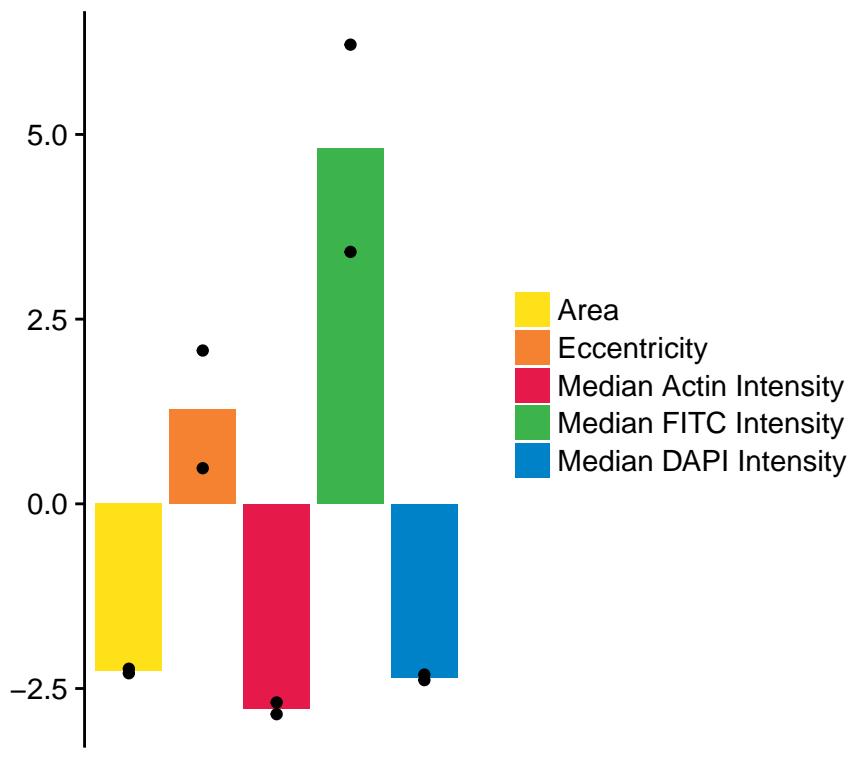


Drug-Induced Phenotypes

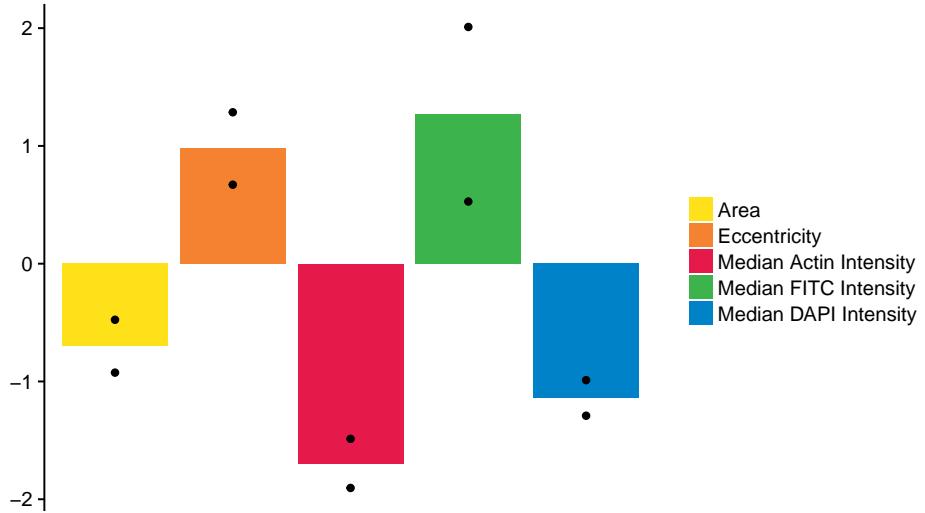


Drug-Induced Phenotypes

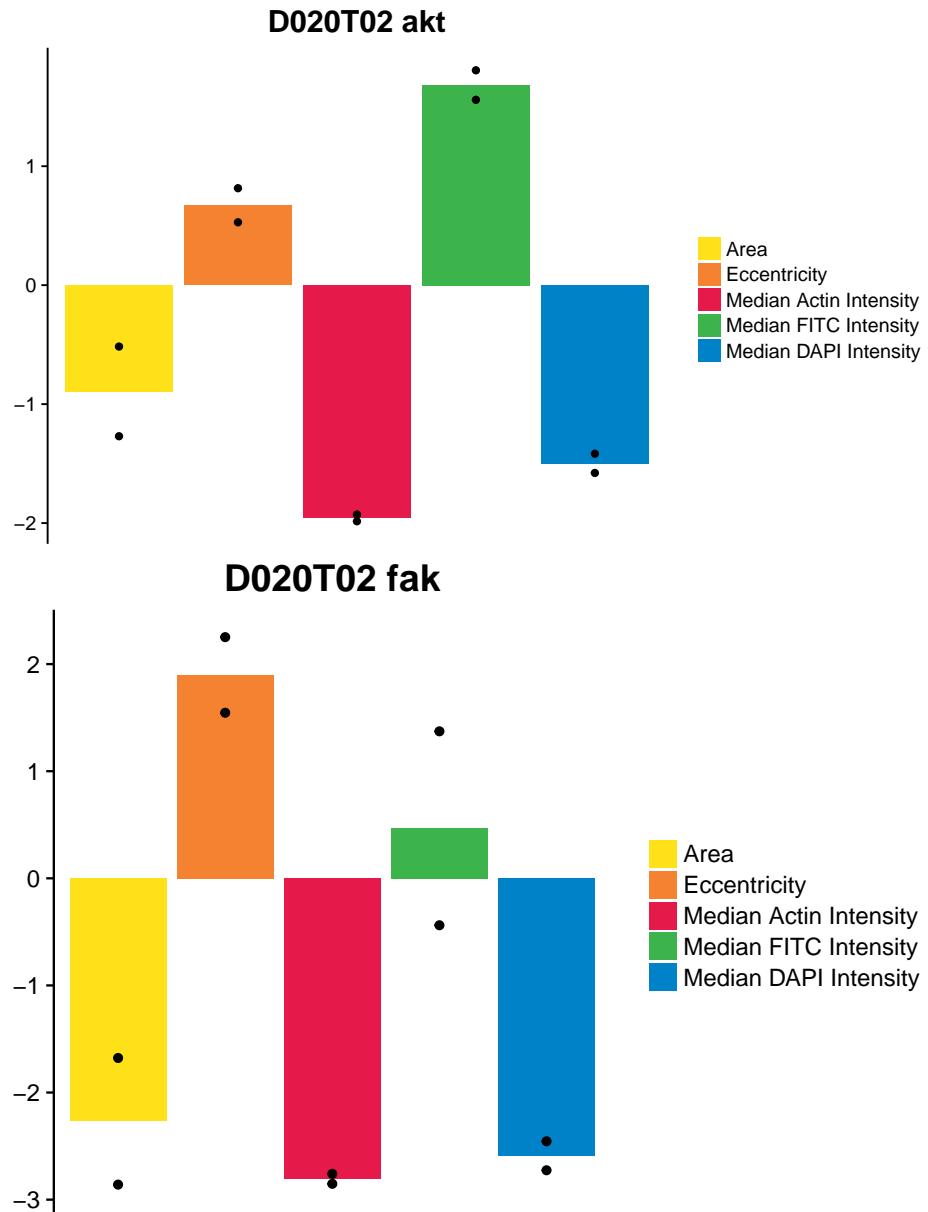
D020T02 her2



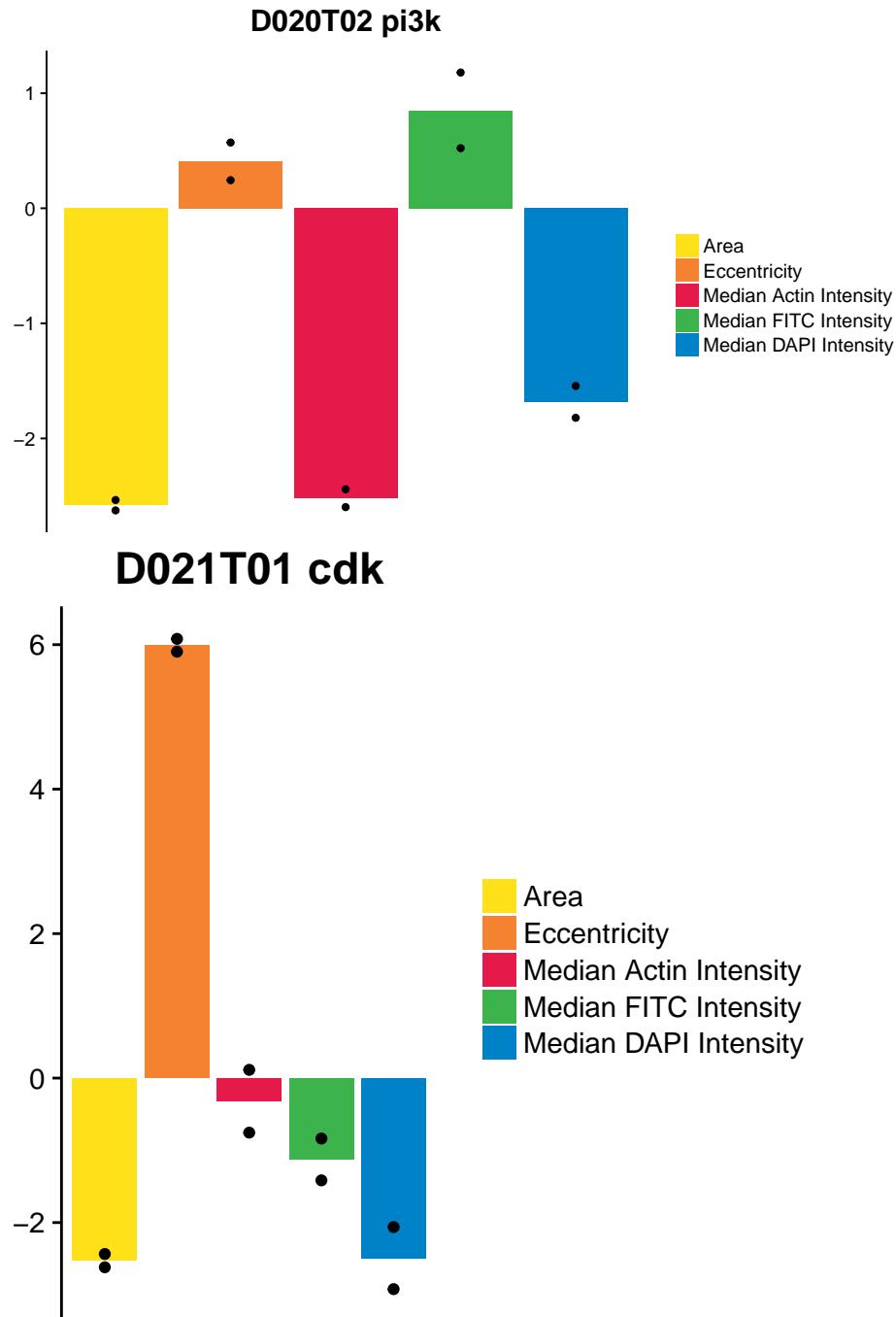
D020T02 flt



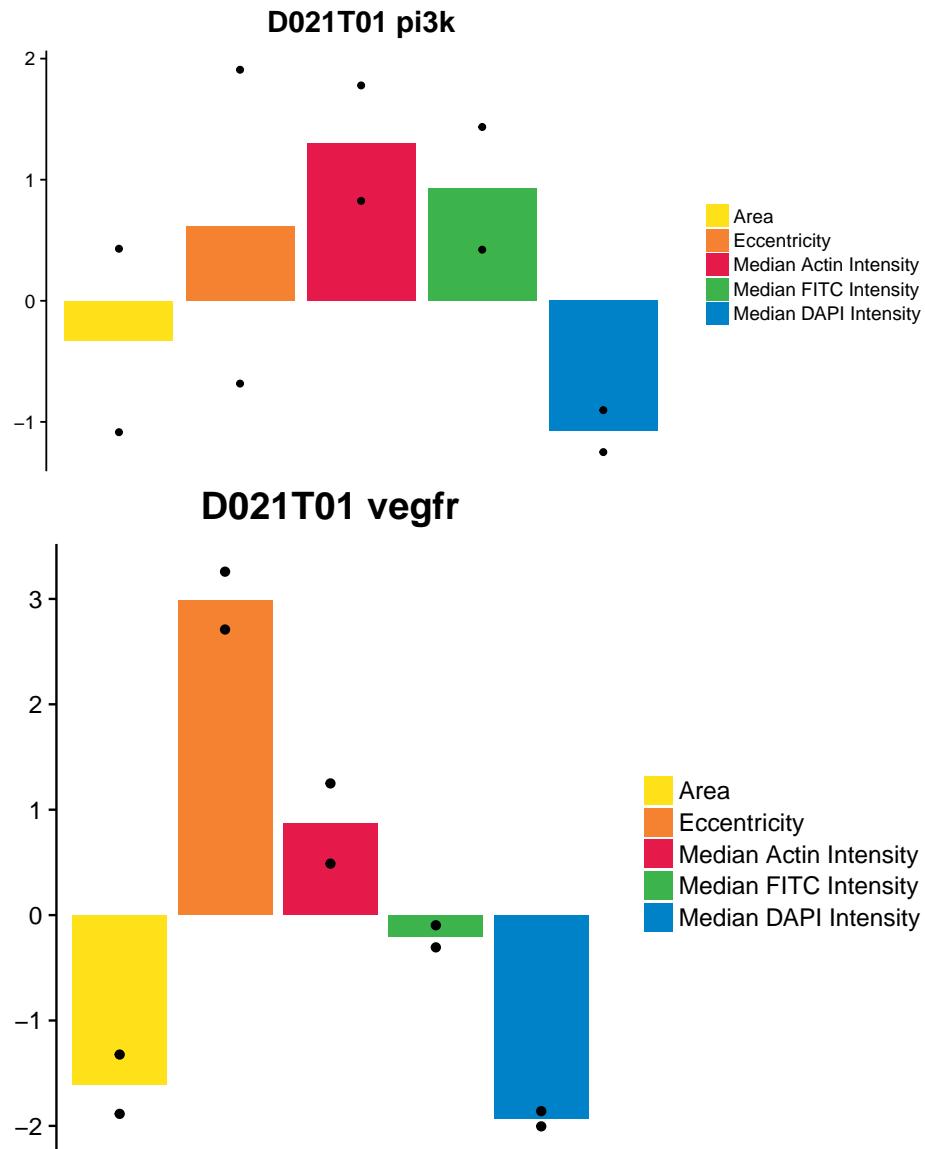
Drug-Induced Phenotypes



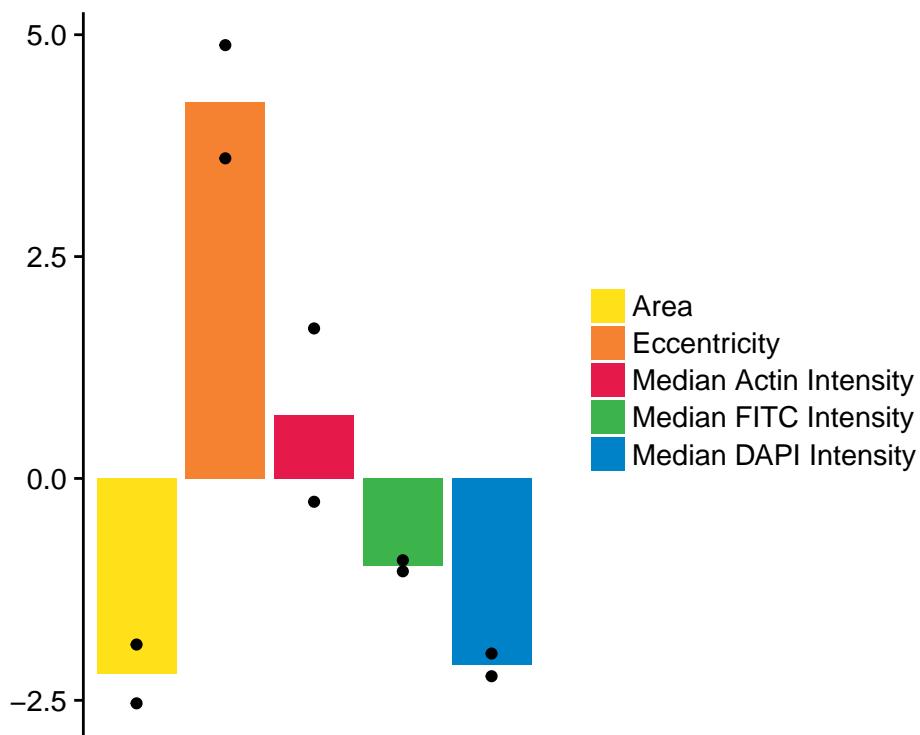
Drug-Induced Phenotypes



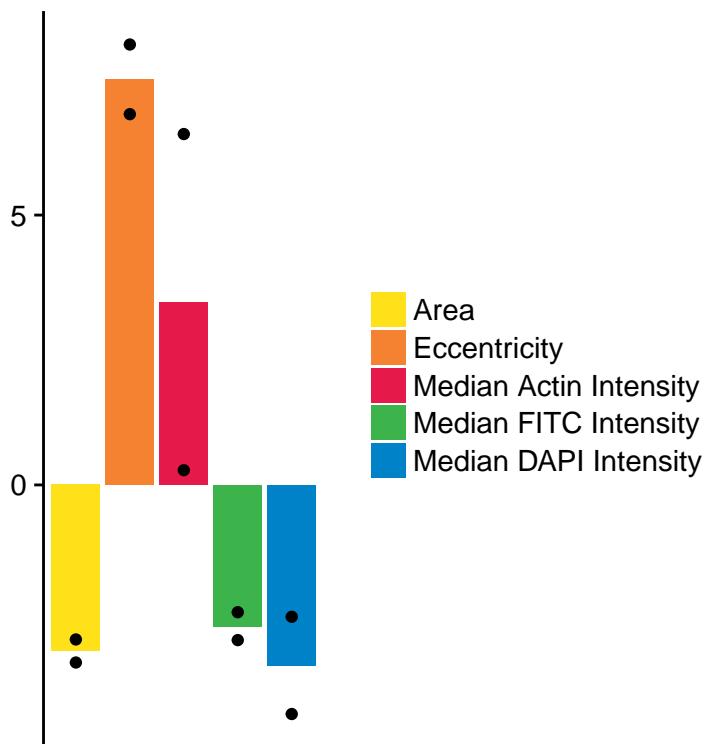
Drug-Induced Phenotypes



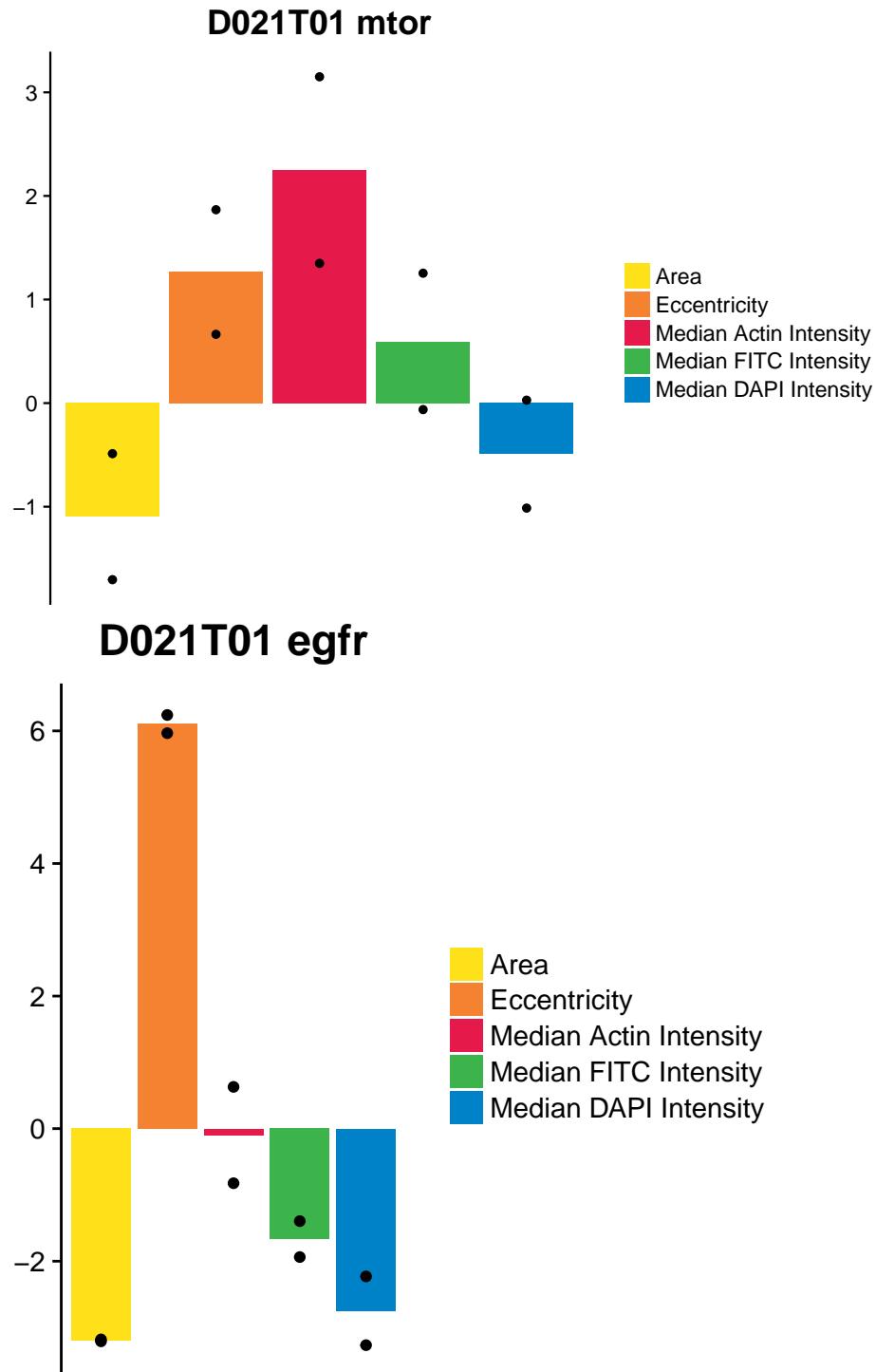
D021T01 aurora kinase



D021T01 alk

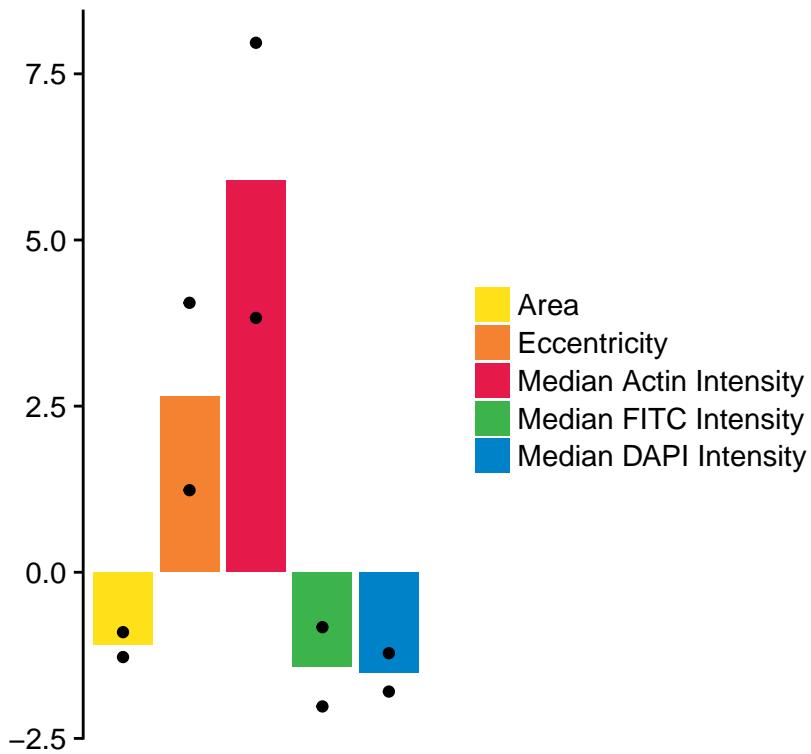


Drug-Induced Phenotypes

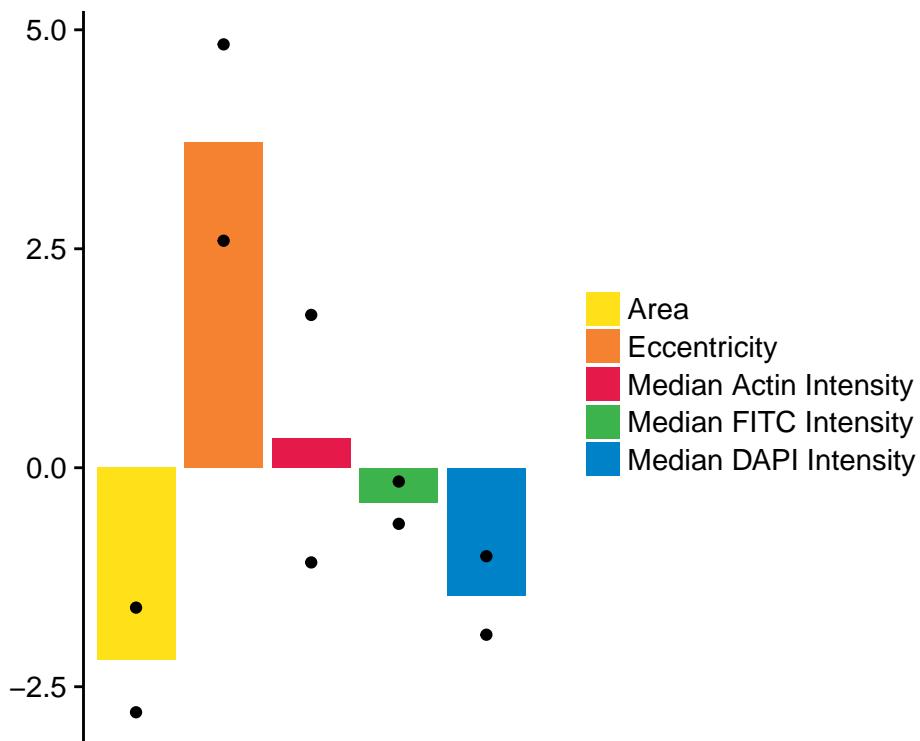


Drug-Induced Phenotypes

D021T01 pdgfr

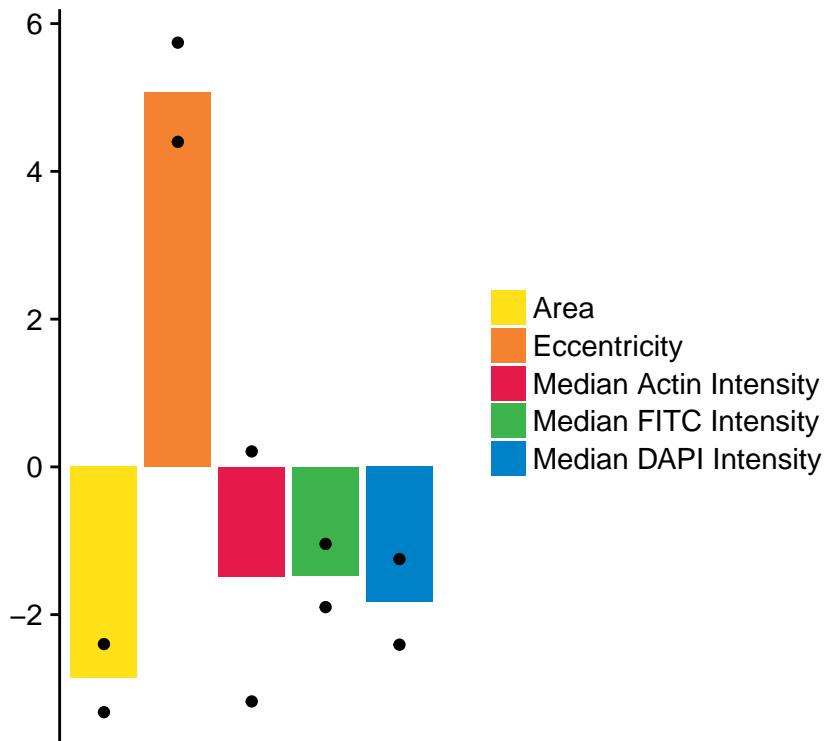


D021T01 bcr-abl

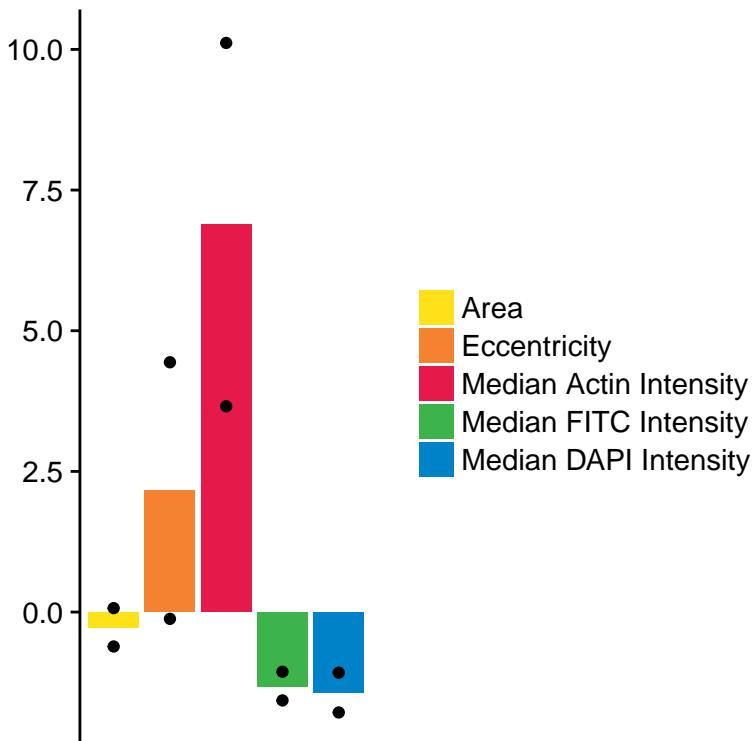


Drug-Induced Phenotypes

D021T01 mek

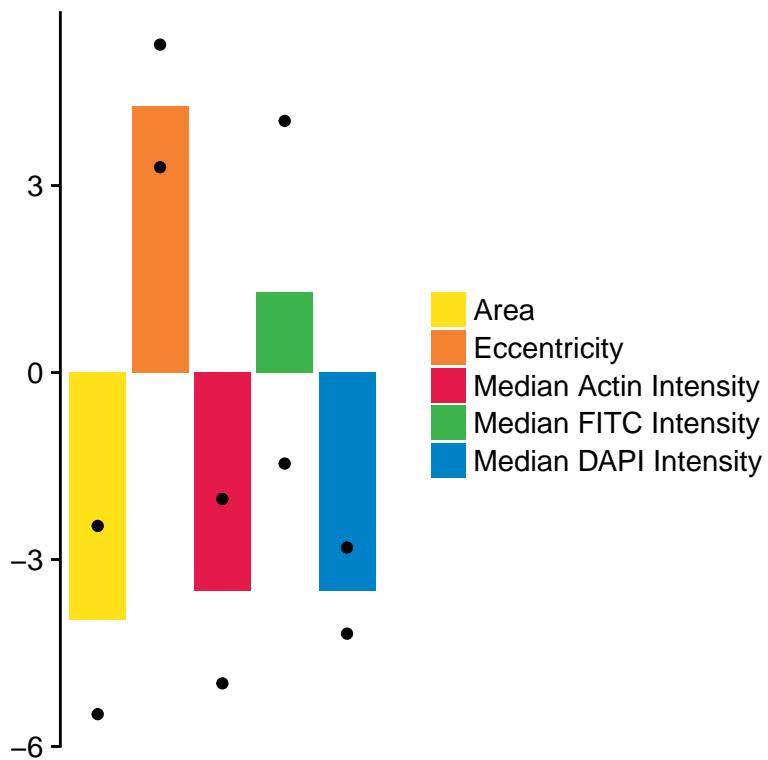


D021T01 c-kit

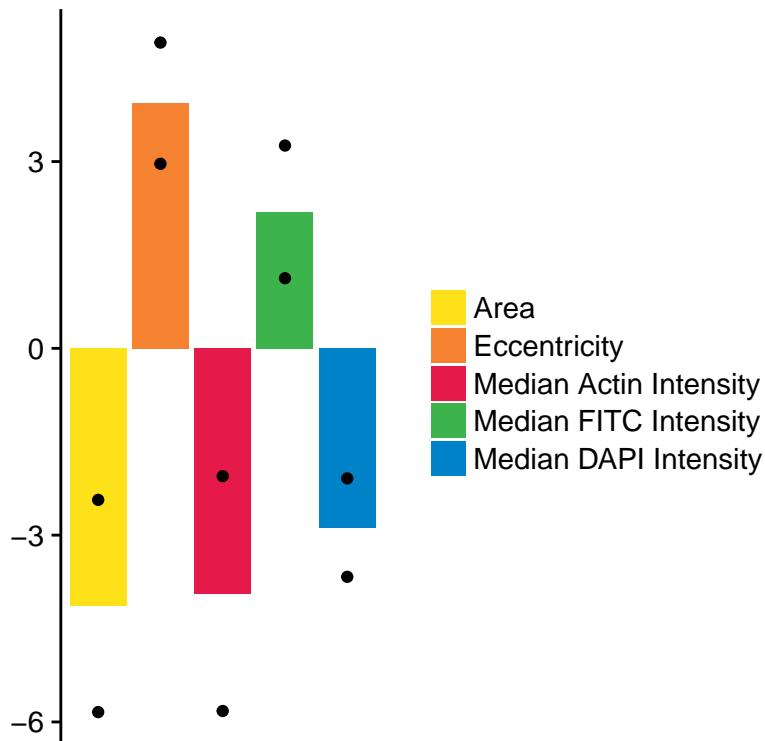


Drug-Induced Phenotypes

D022T01 cdk

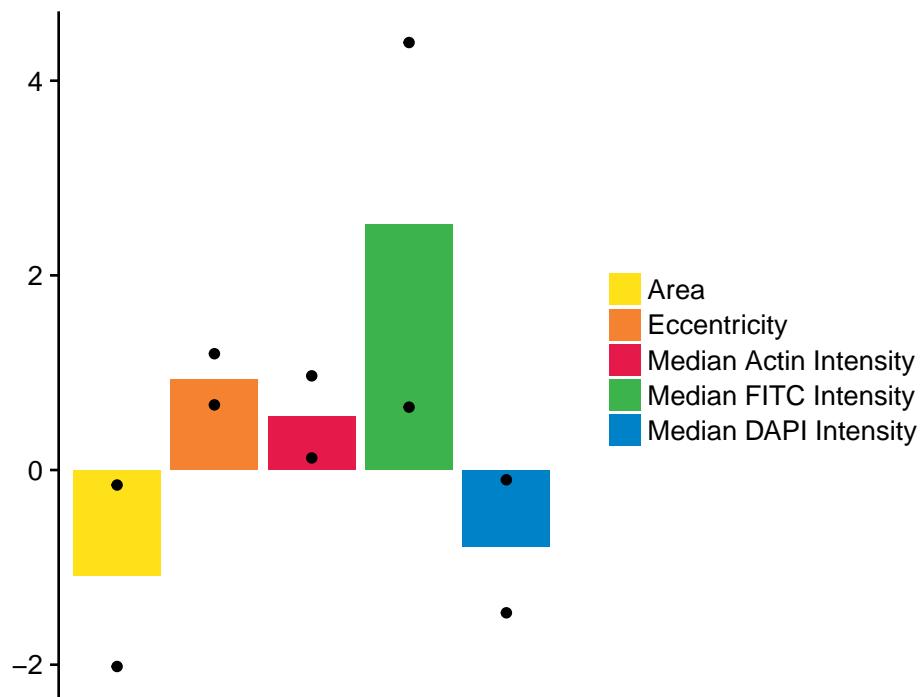


D022T01 aurora kinase

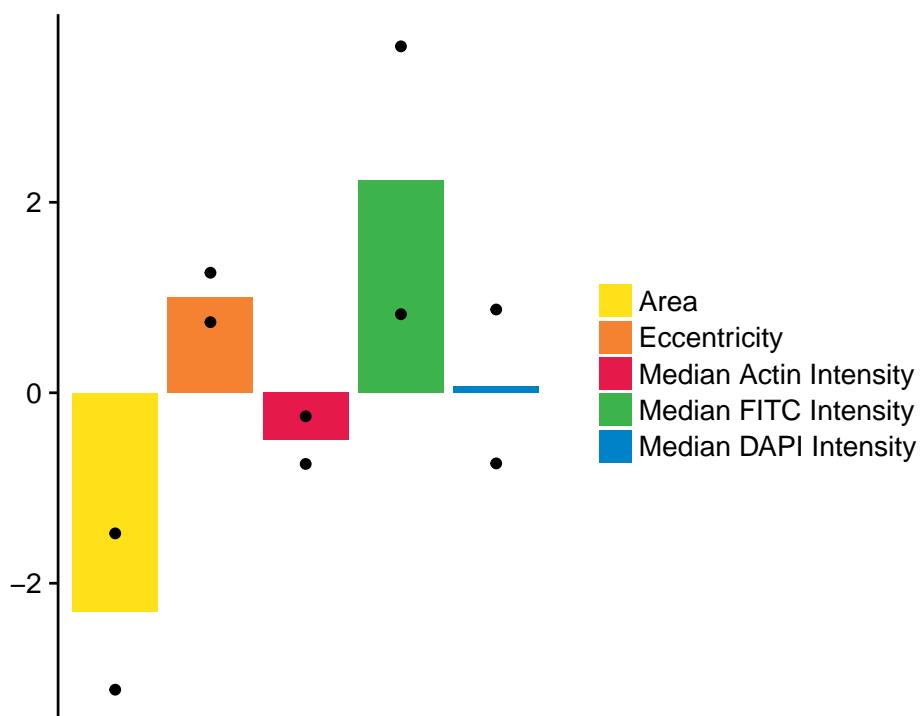


Drug-Induced Phenotypes

D022T01 akt

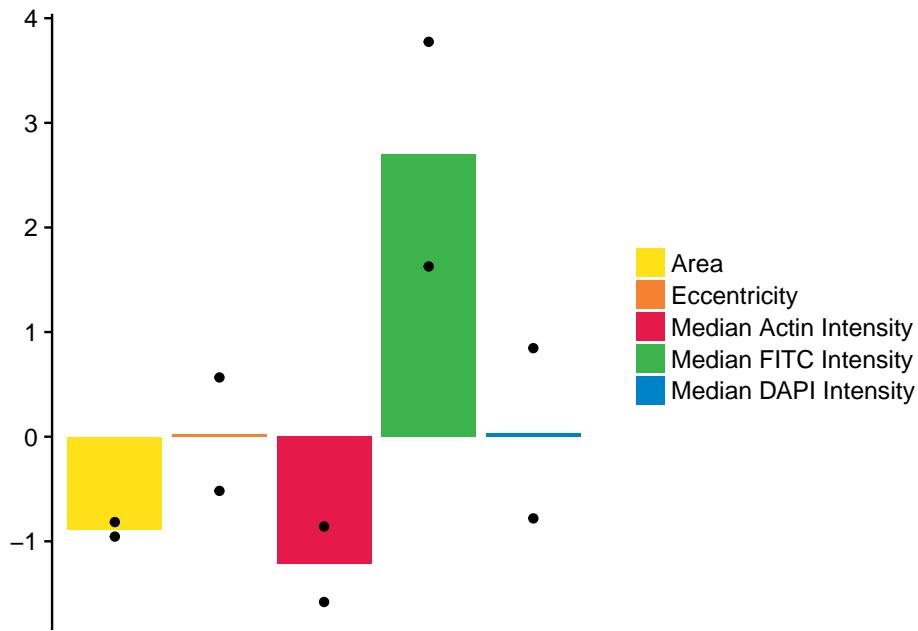


D022T01 mtor

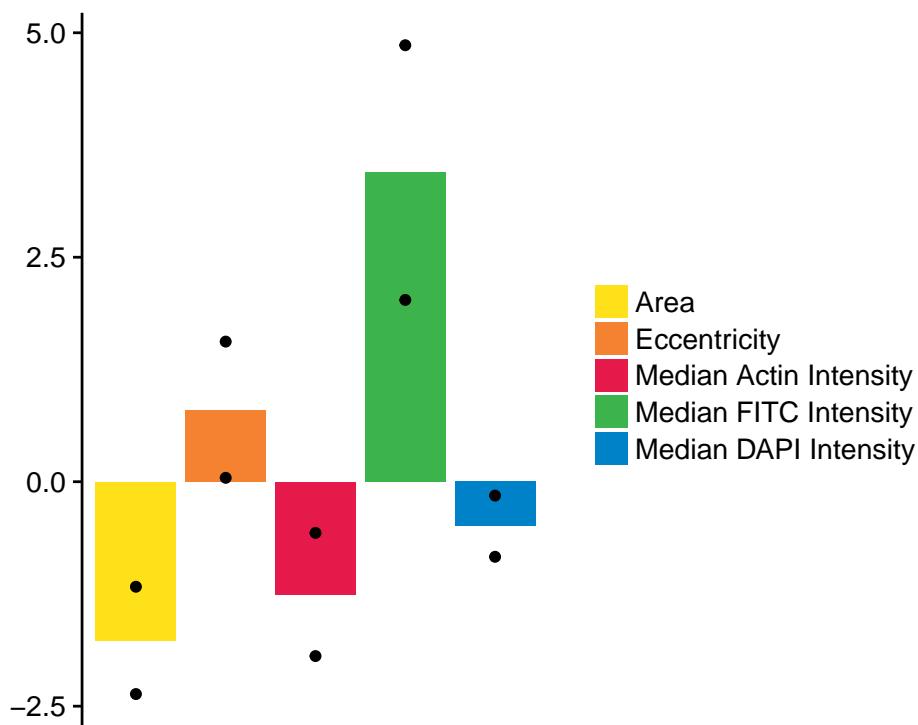


Drug-Induced Phenotypes

D022T01 egfr

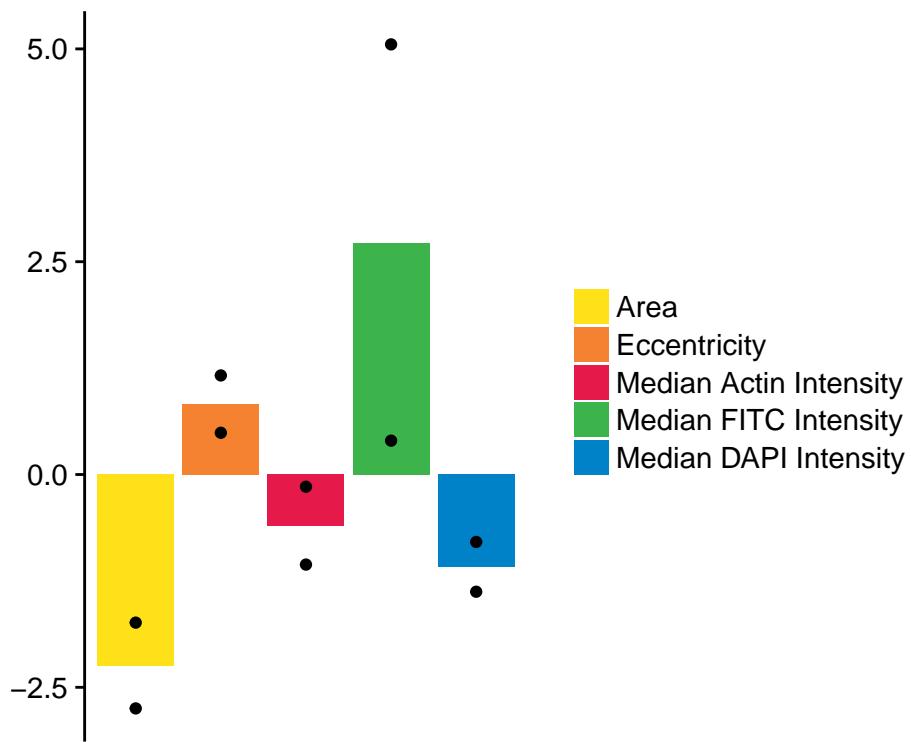


D022T01 mek

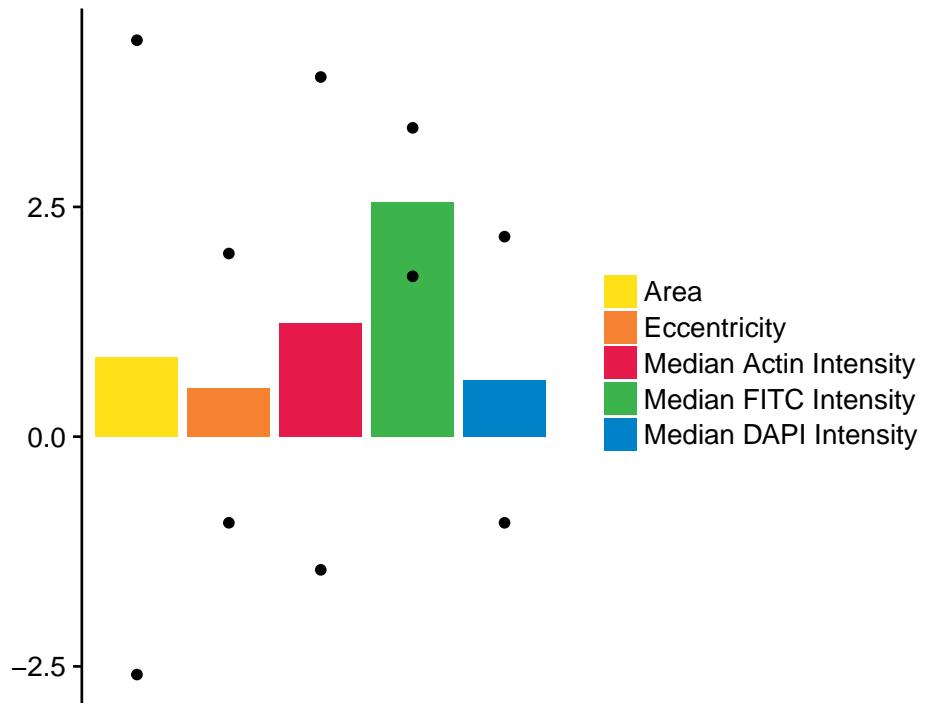


Drug-Induced Phenotypes

D022T01 pi3k

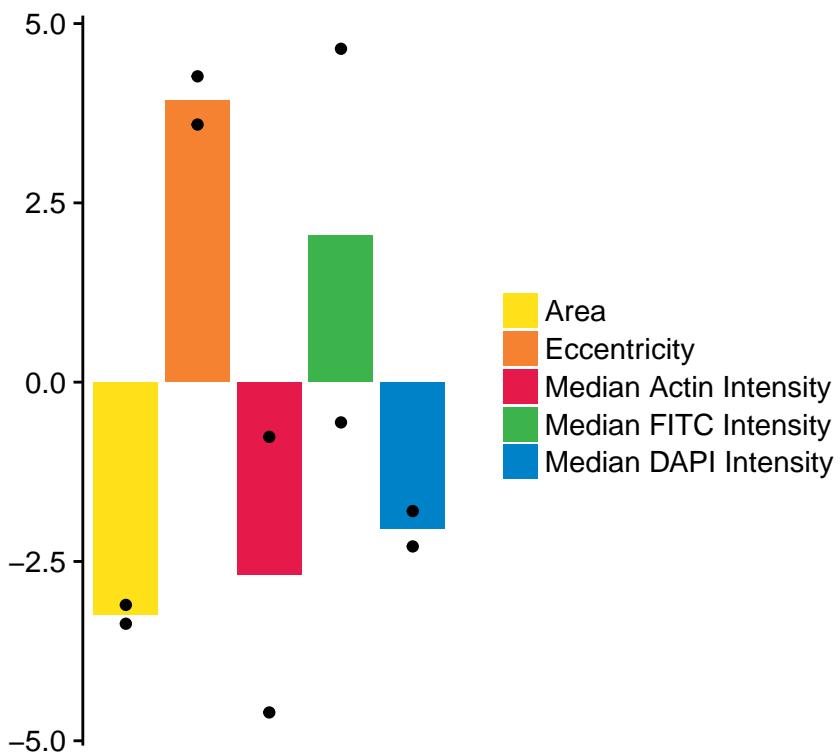


D027T01 cdk

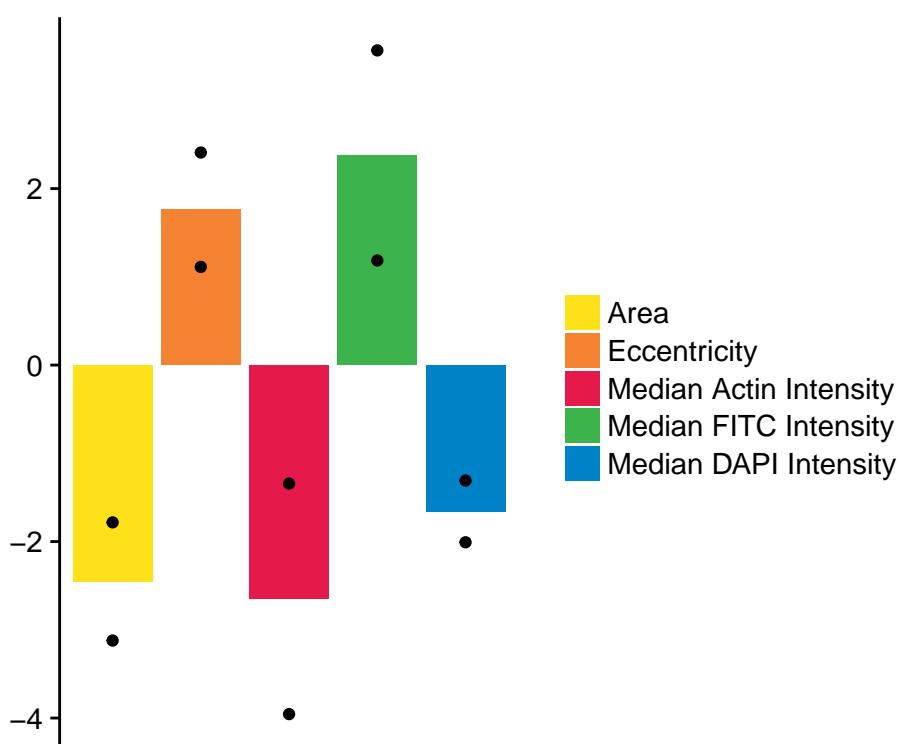


Drug-Induced Phenotypes

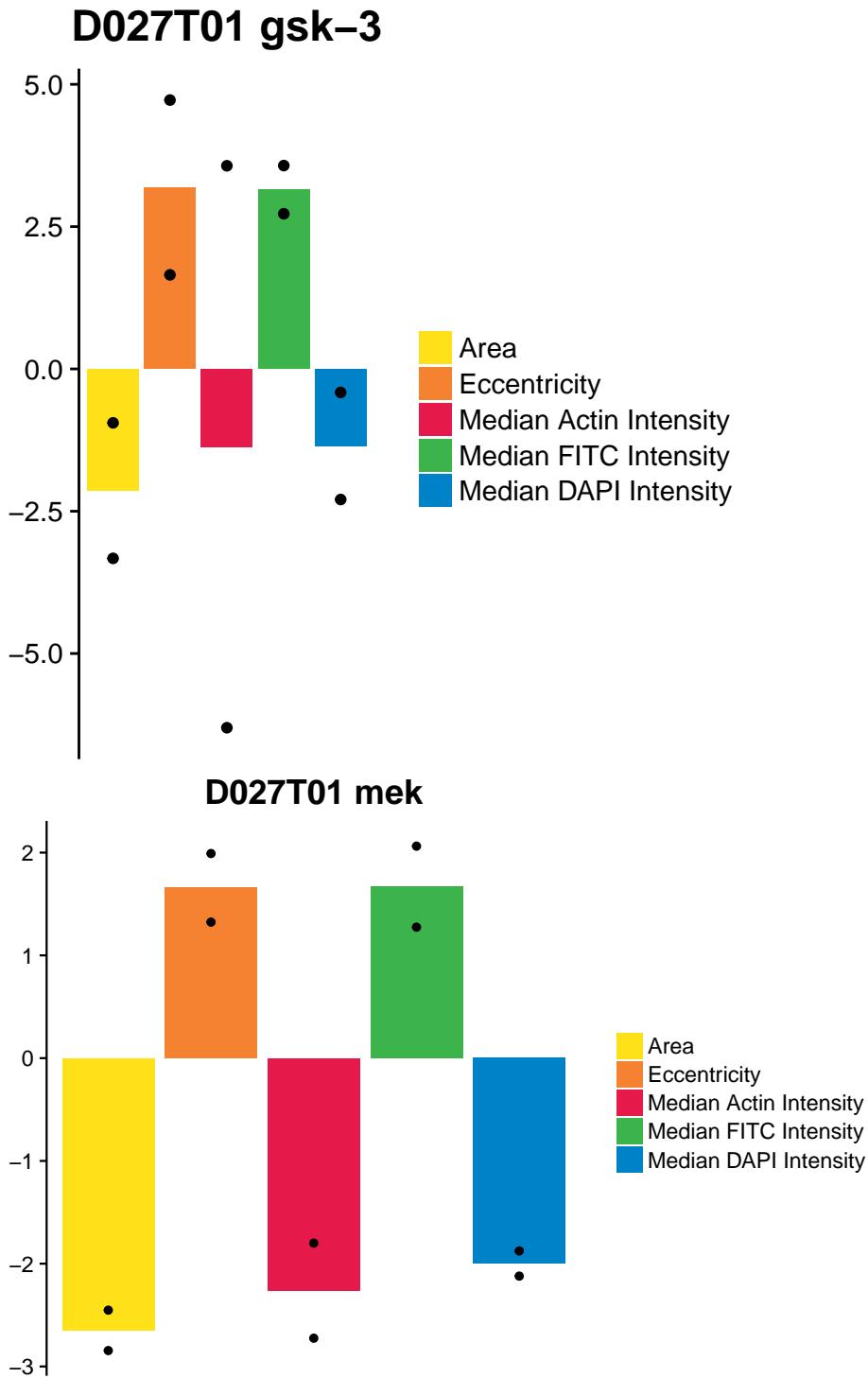
D027T01 vegfr



D027T01 egfr

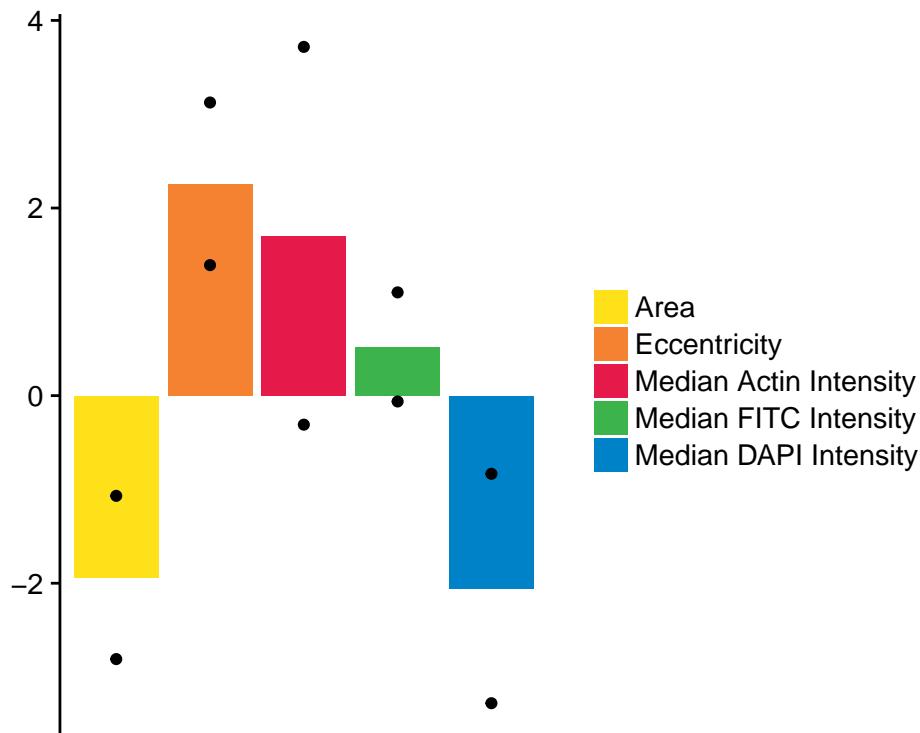


Drug-Induced Phenotypes

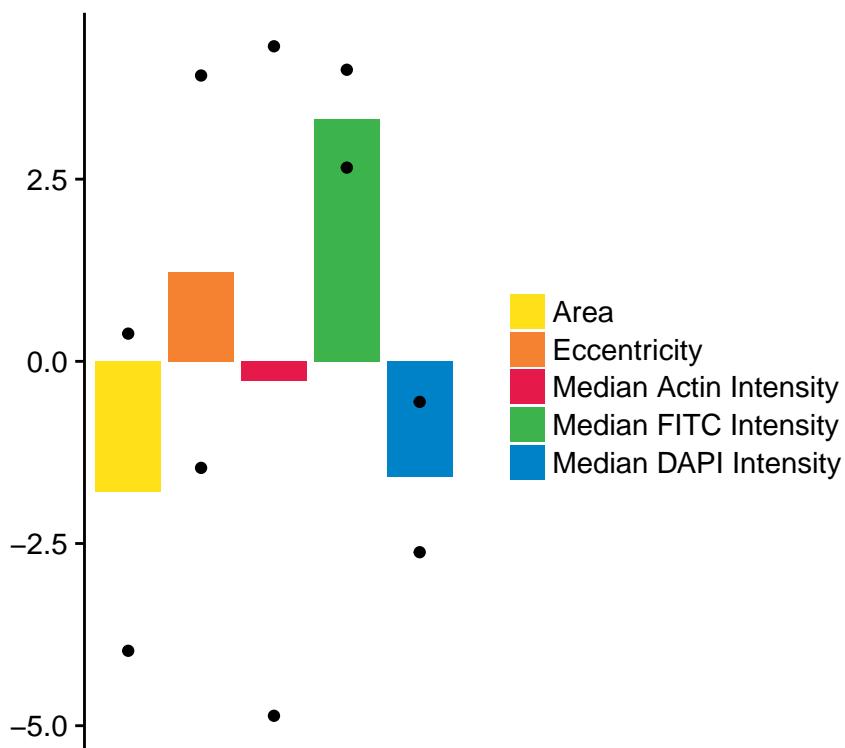


Drug-Induced Phenotypes

D027T01 fak

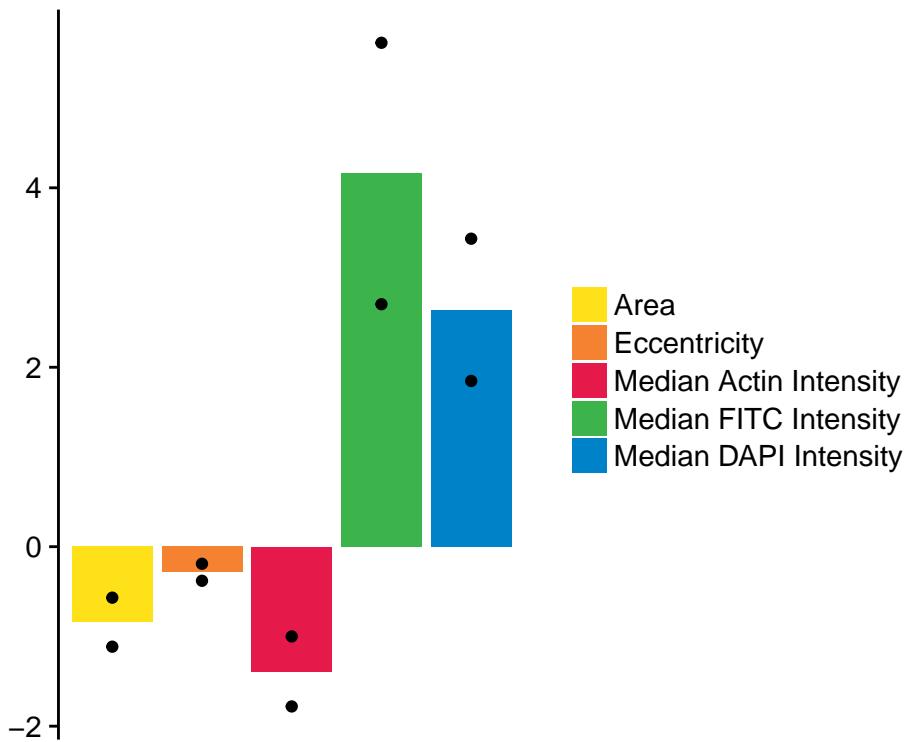


D030T01 cdk

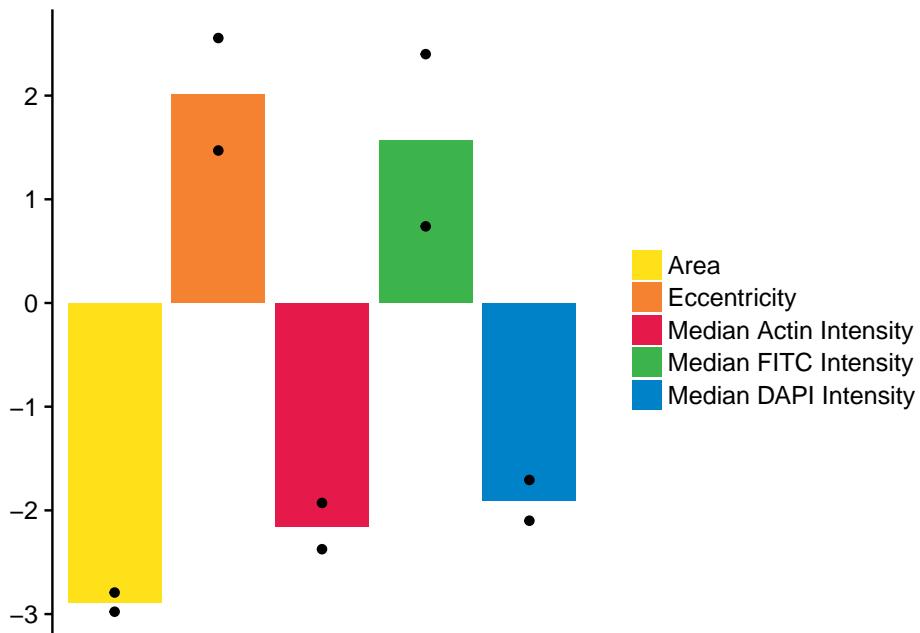


Drug-Induced Phenotypes

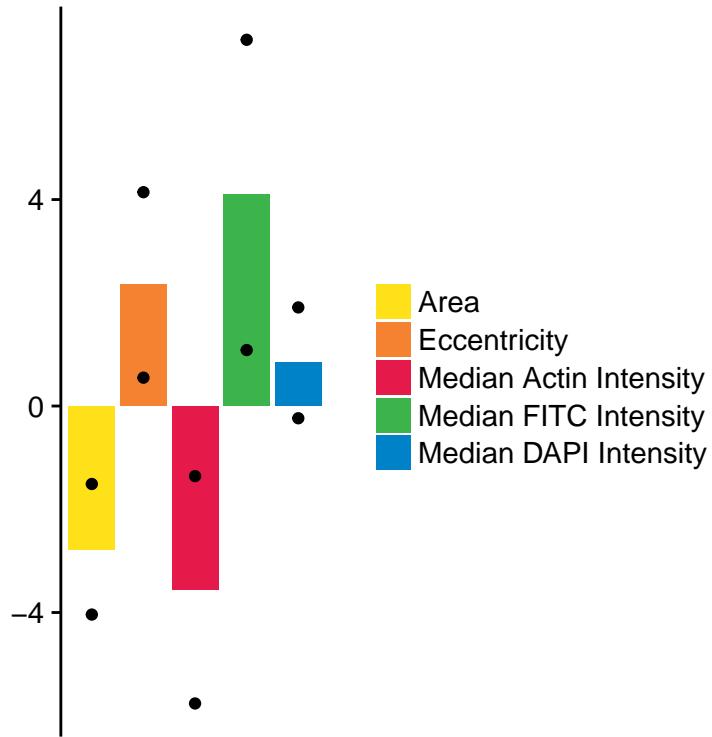
D030T01 egfr



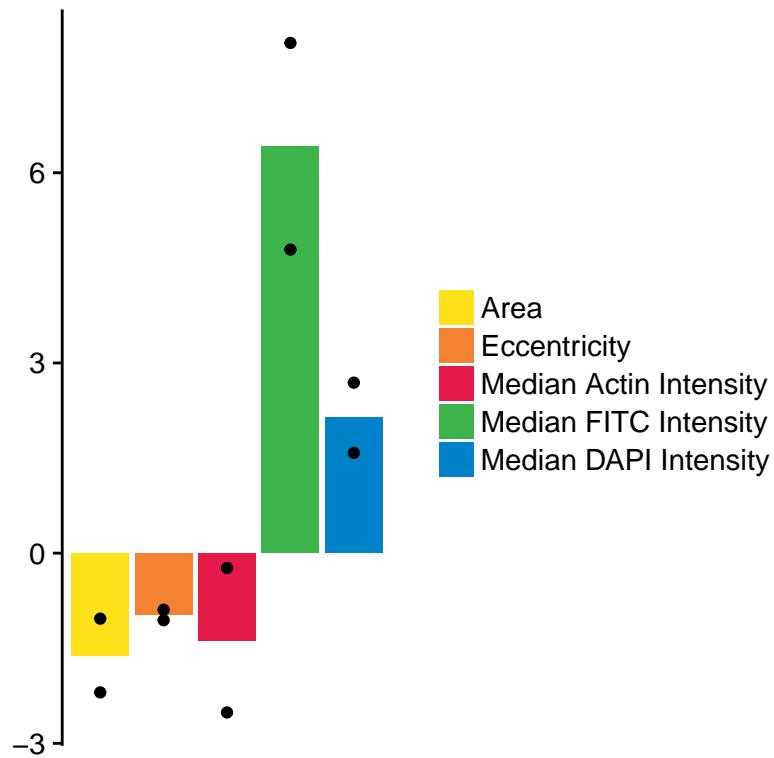
D030T01 aurora kinase



D030T01 gsk-3

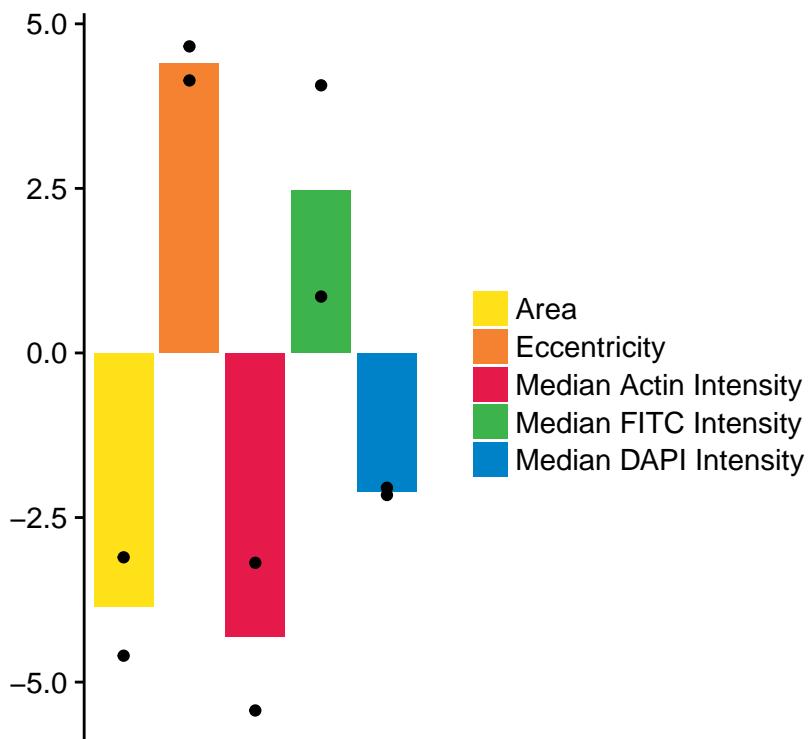


D030T01 mtor

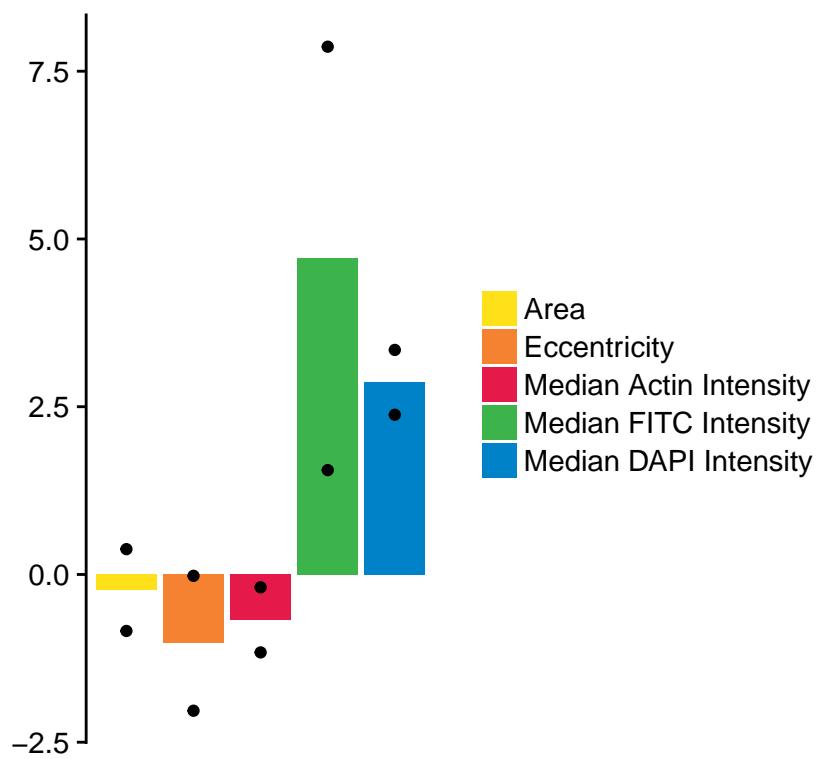


Drug-Induced Phenotypes

D030T01 plk

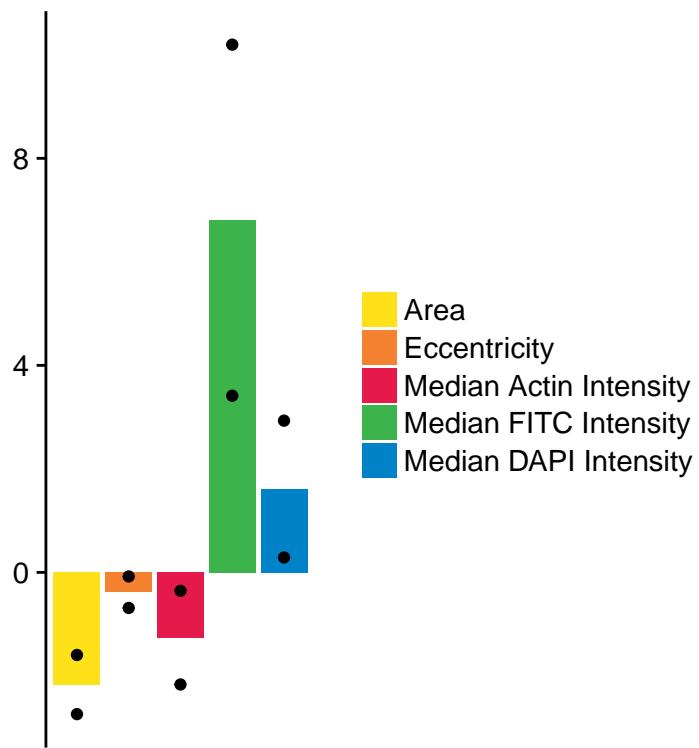


D030T01 pi3k

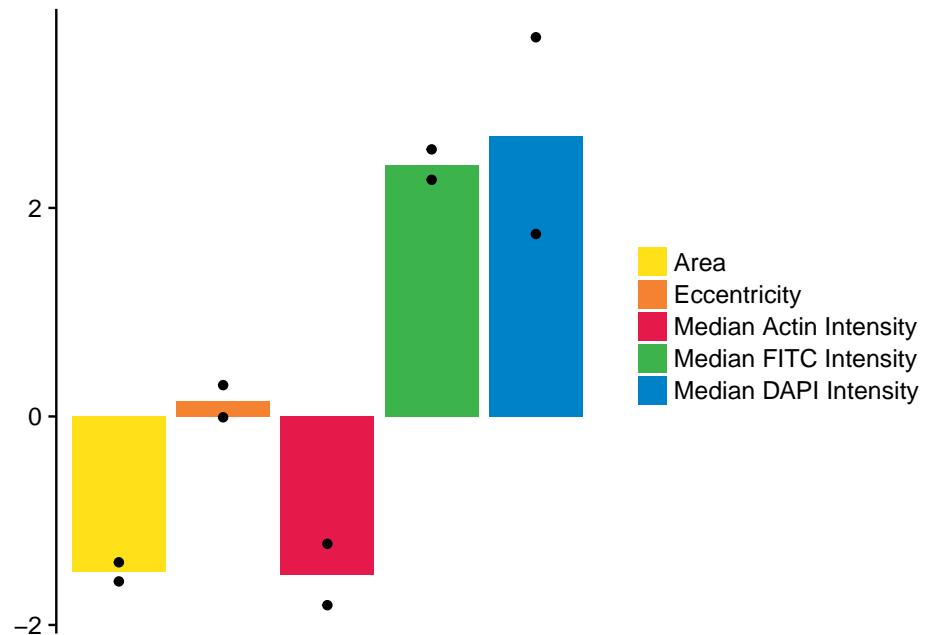


Drug-Induced Phenotypes

D030T01 mek

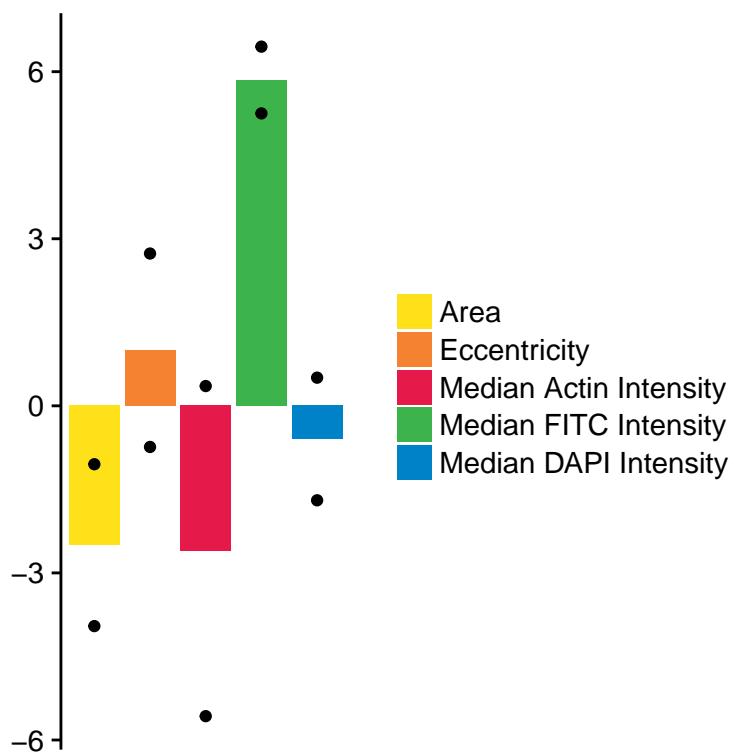


D030T01 bcr-abl

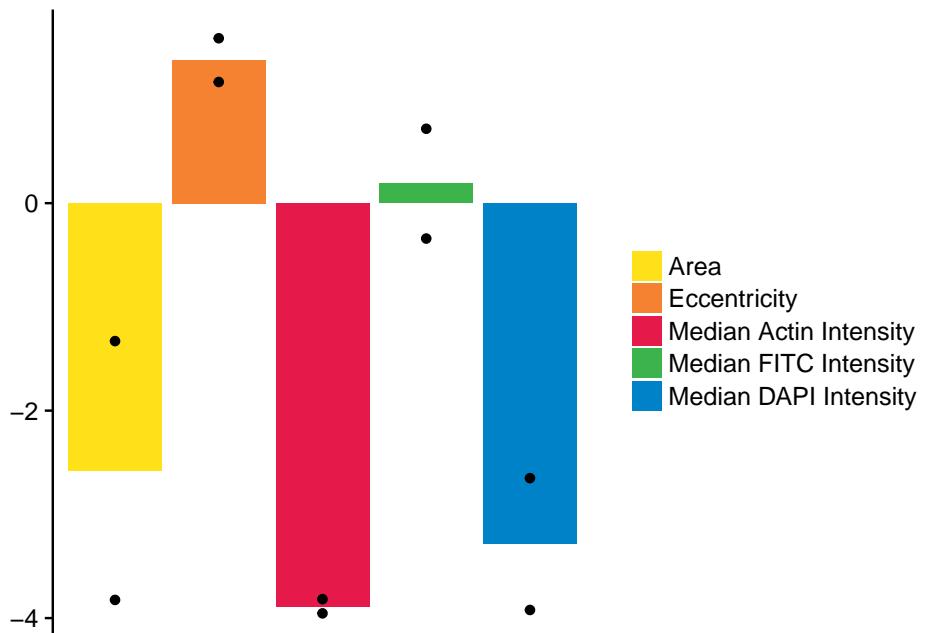


Drug-Induced Phenotypes

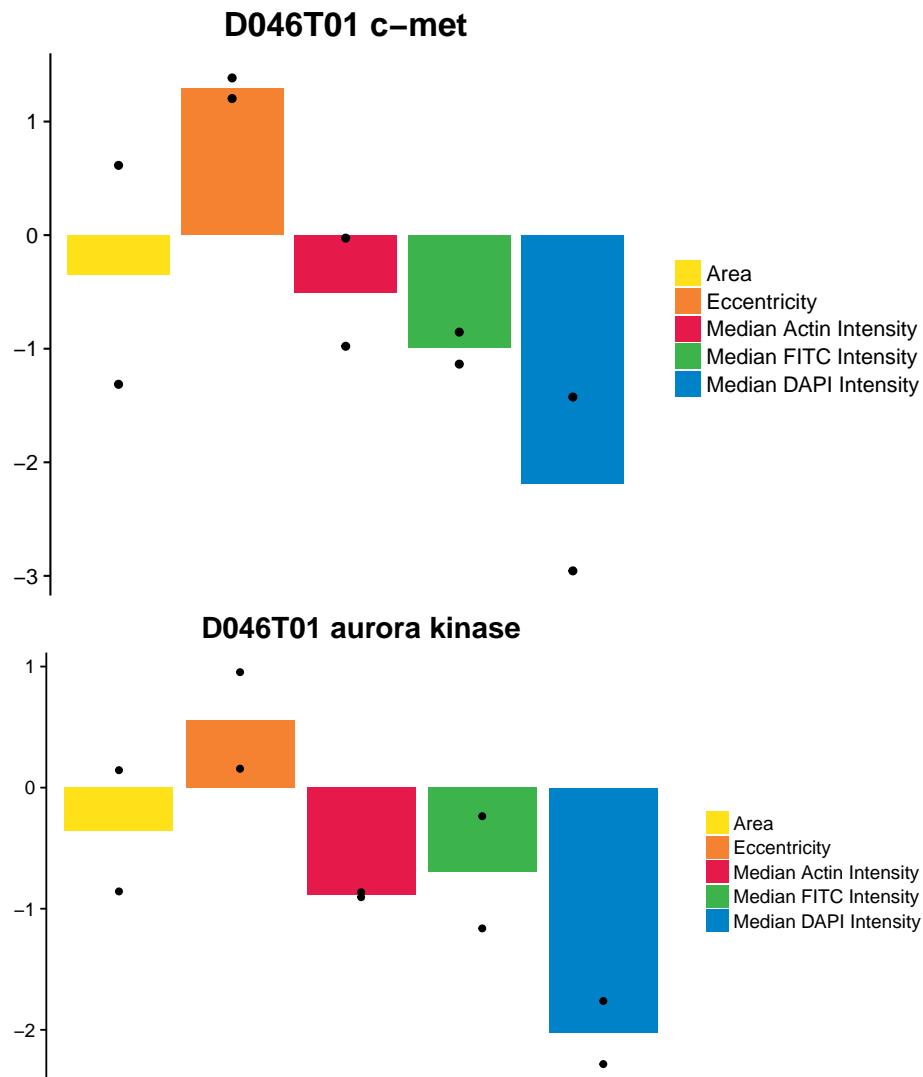
D030T01 fak



D046T01 cdk

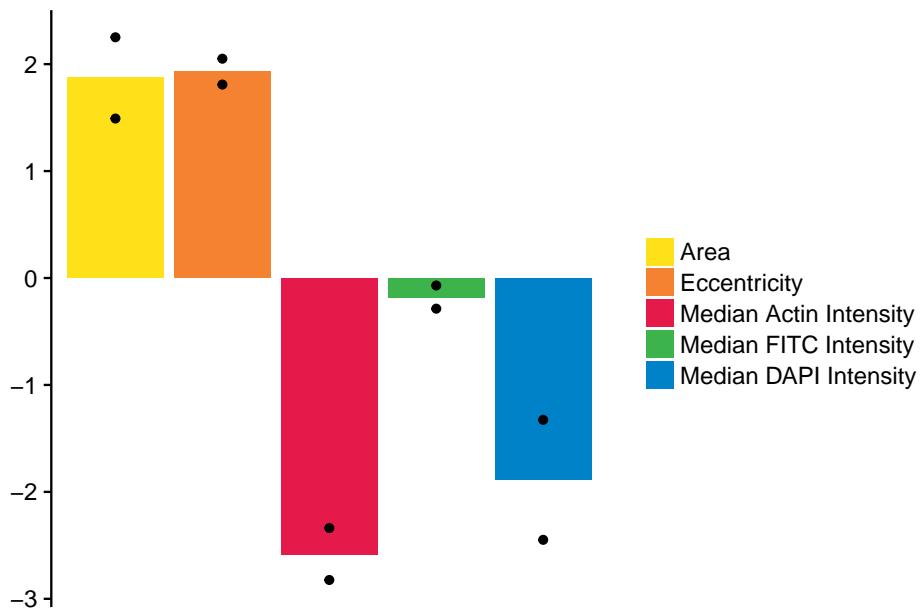


Drug-Induced Phenotypes

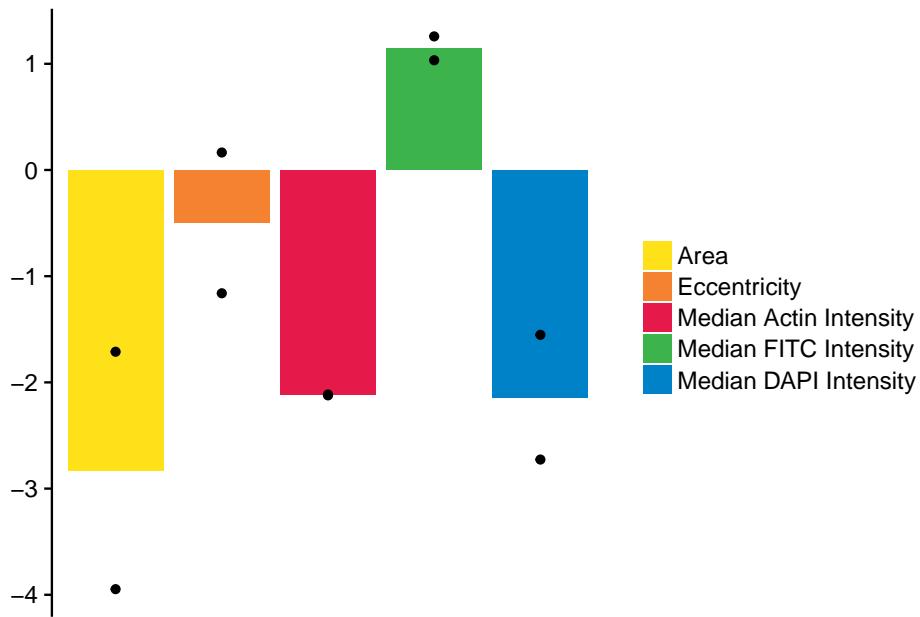


Drug-Induced Phenotypes

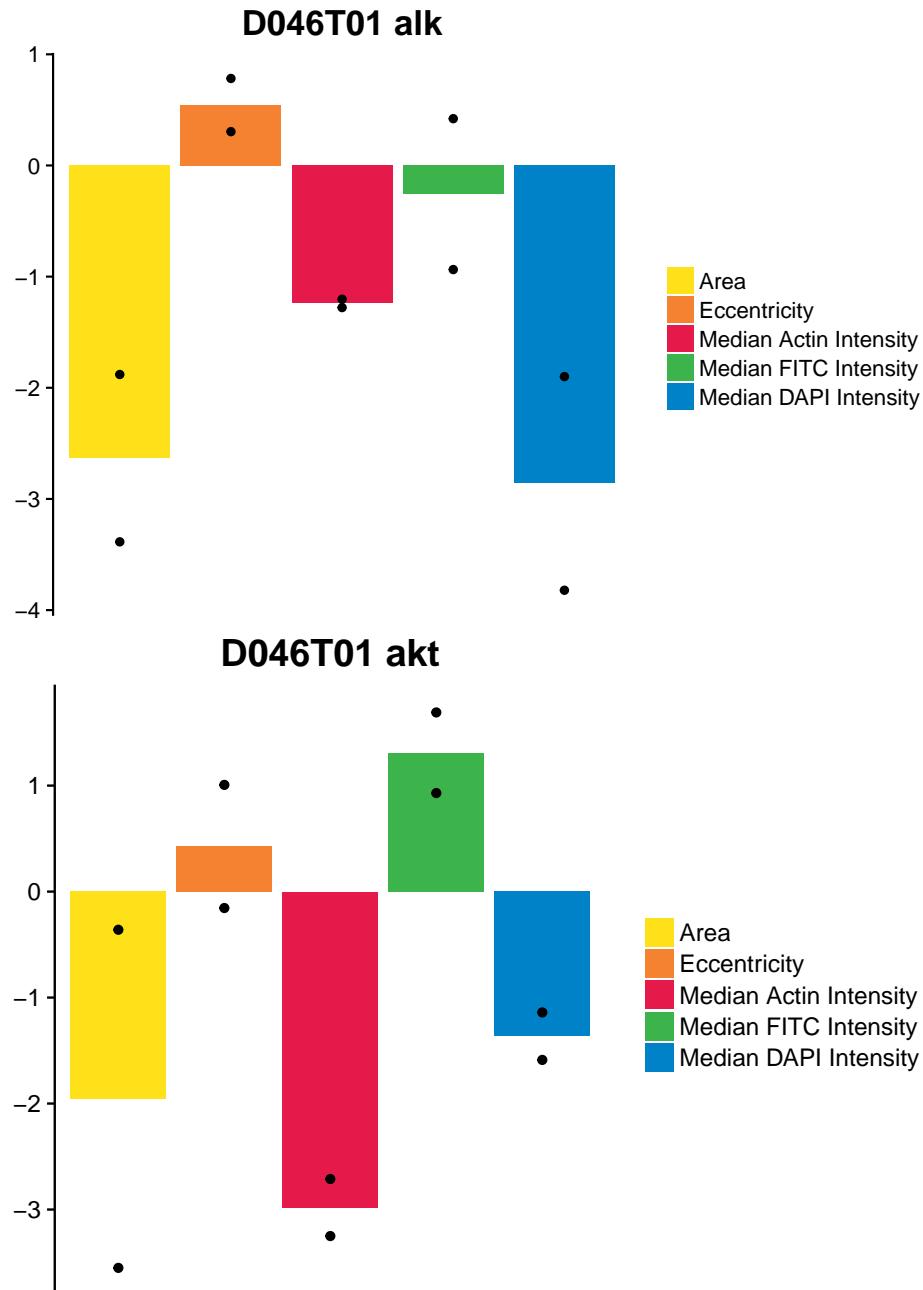
D046T01 gsk-3



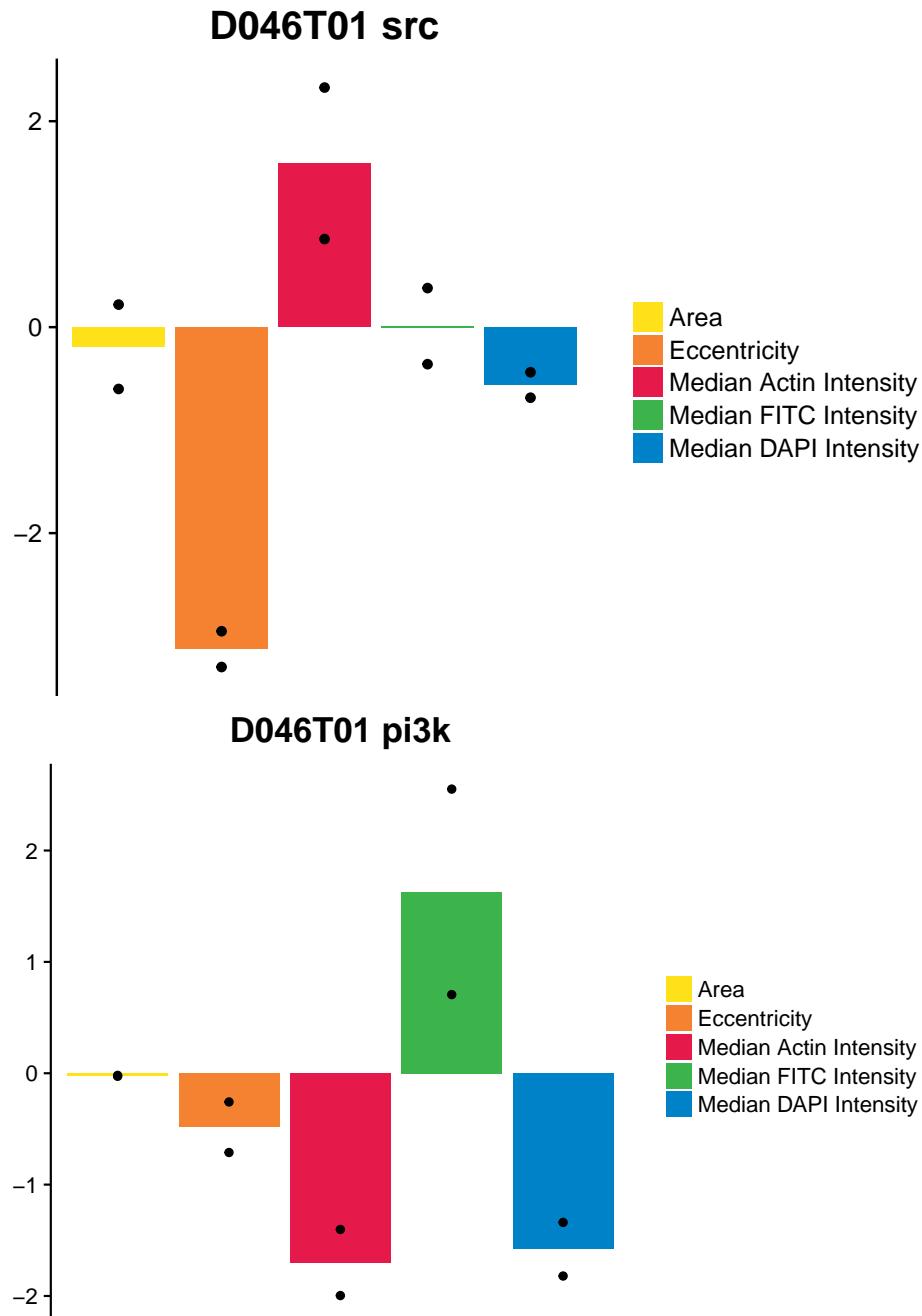
D046T01 mtor



Drug-Induced Phenotypes

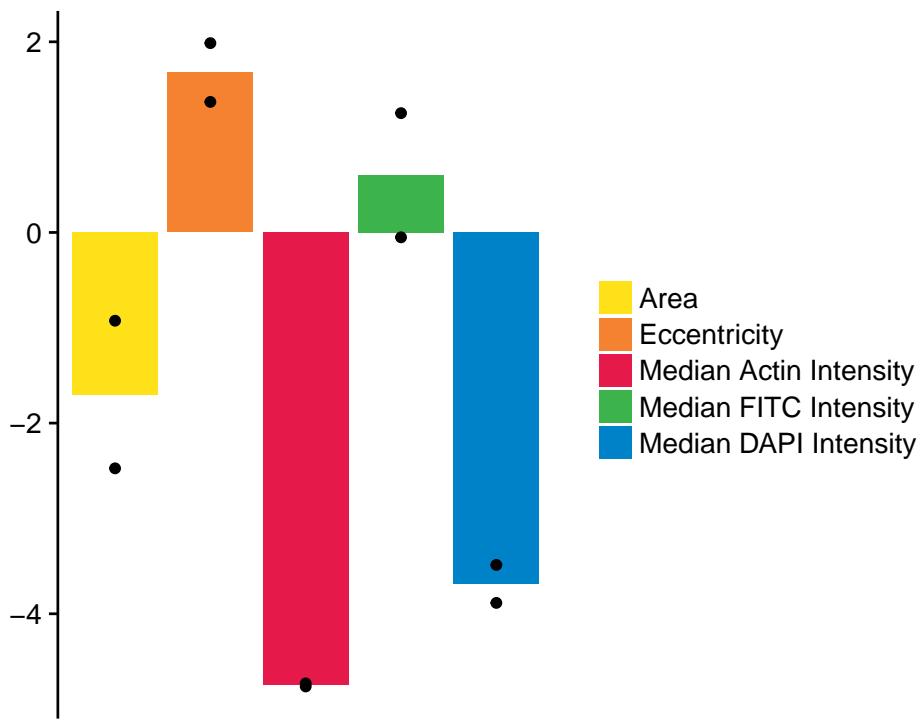


Drug-Induced Phenotypes

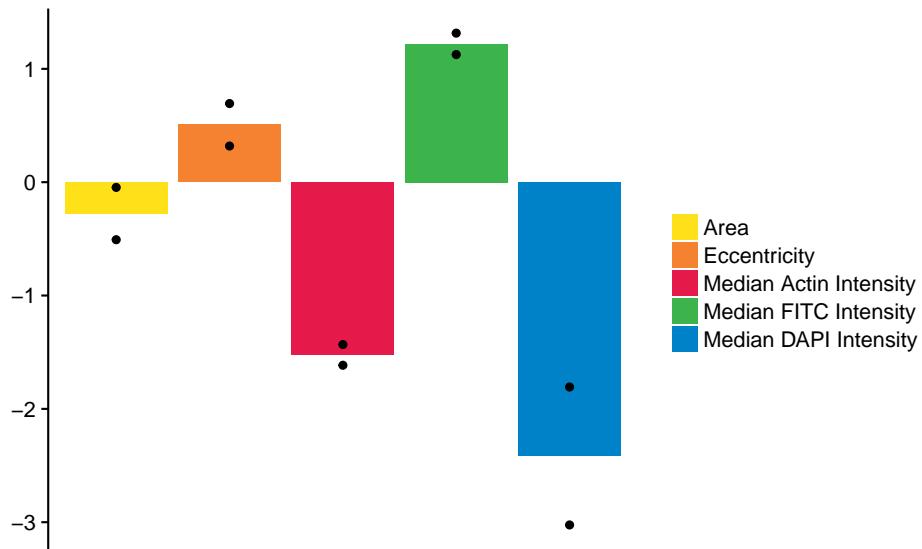


Drug-Induced Phenotypes

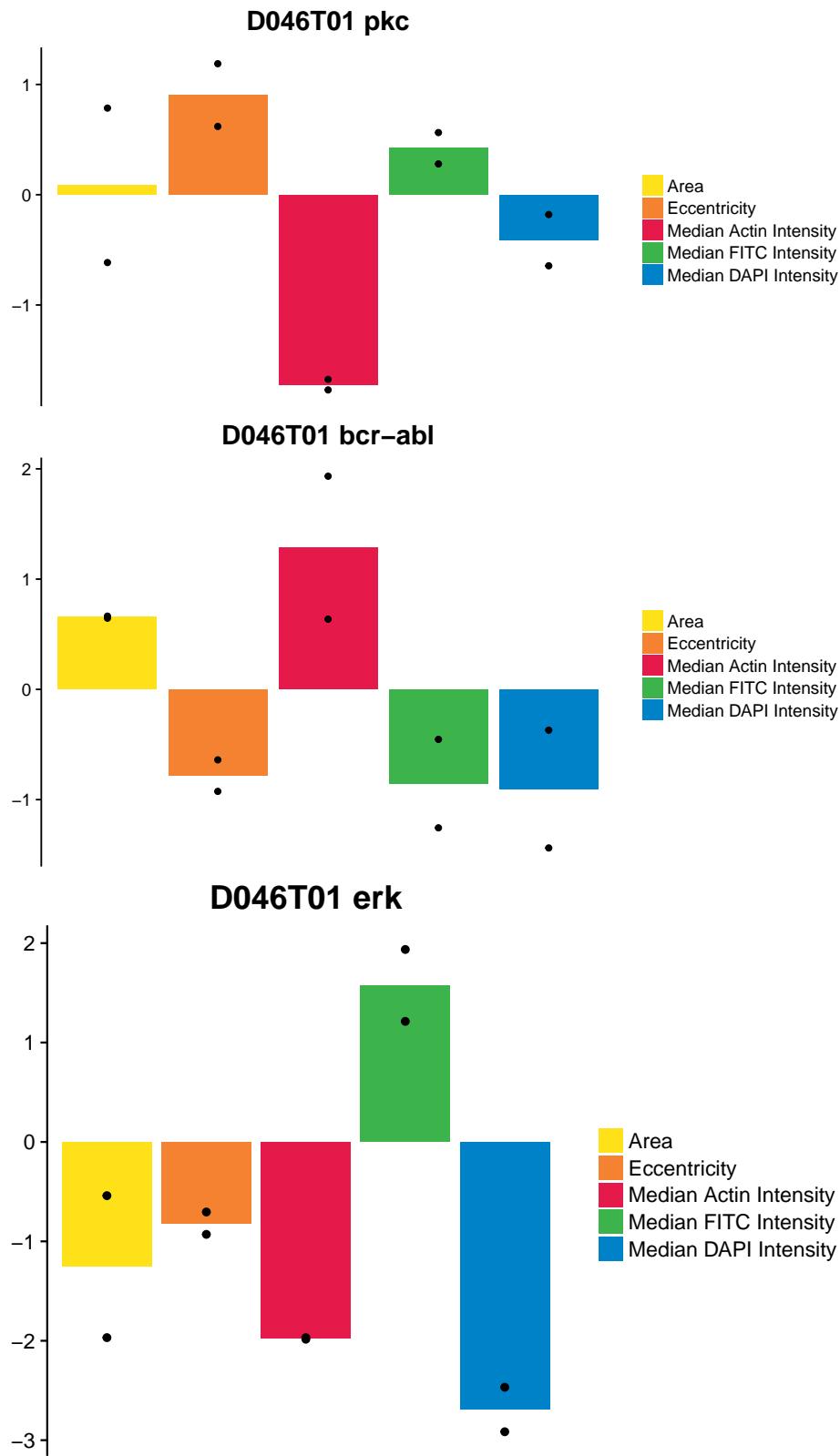
D046T01 mek



D046T01 vegfr

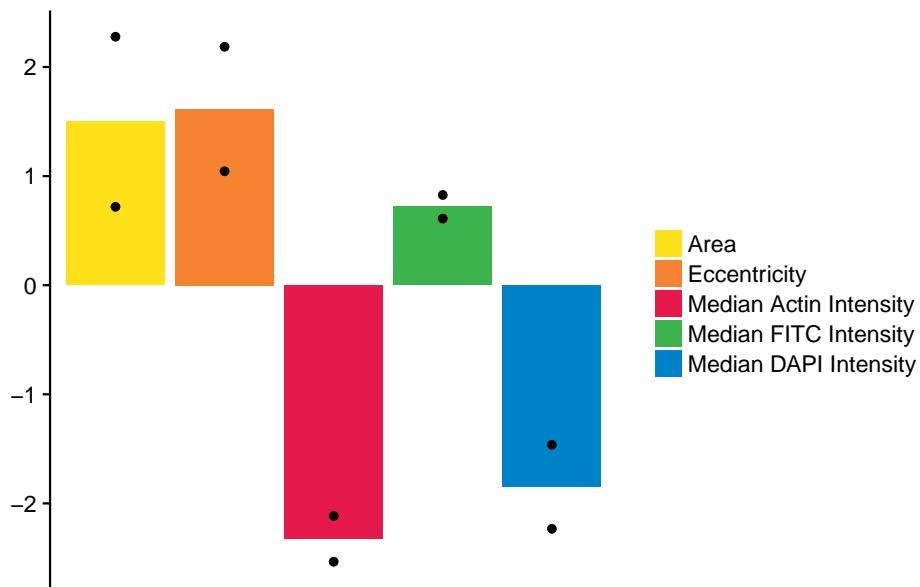


Drug-Induced Phenotypes

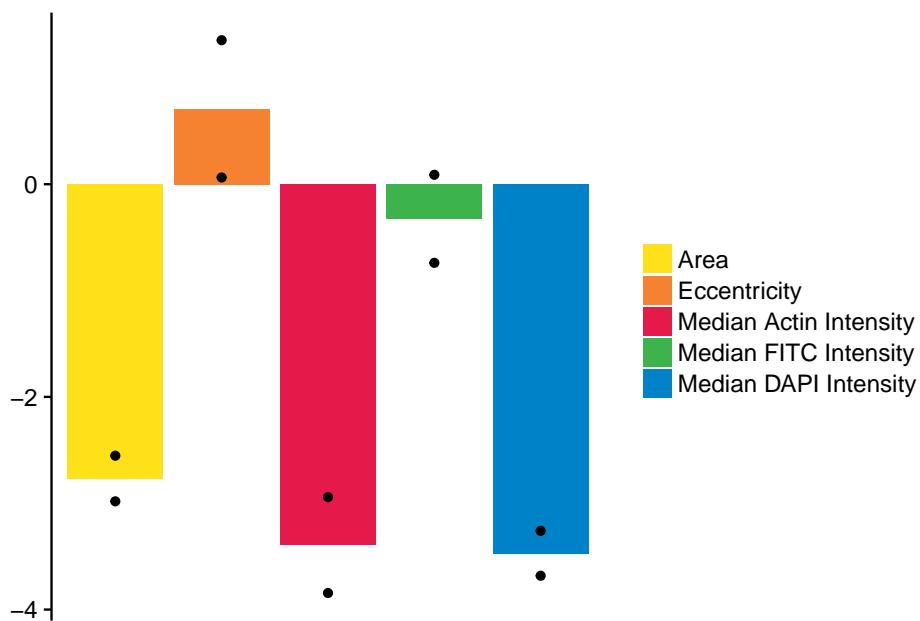


Drug-Induced Phenotypes

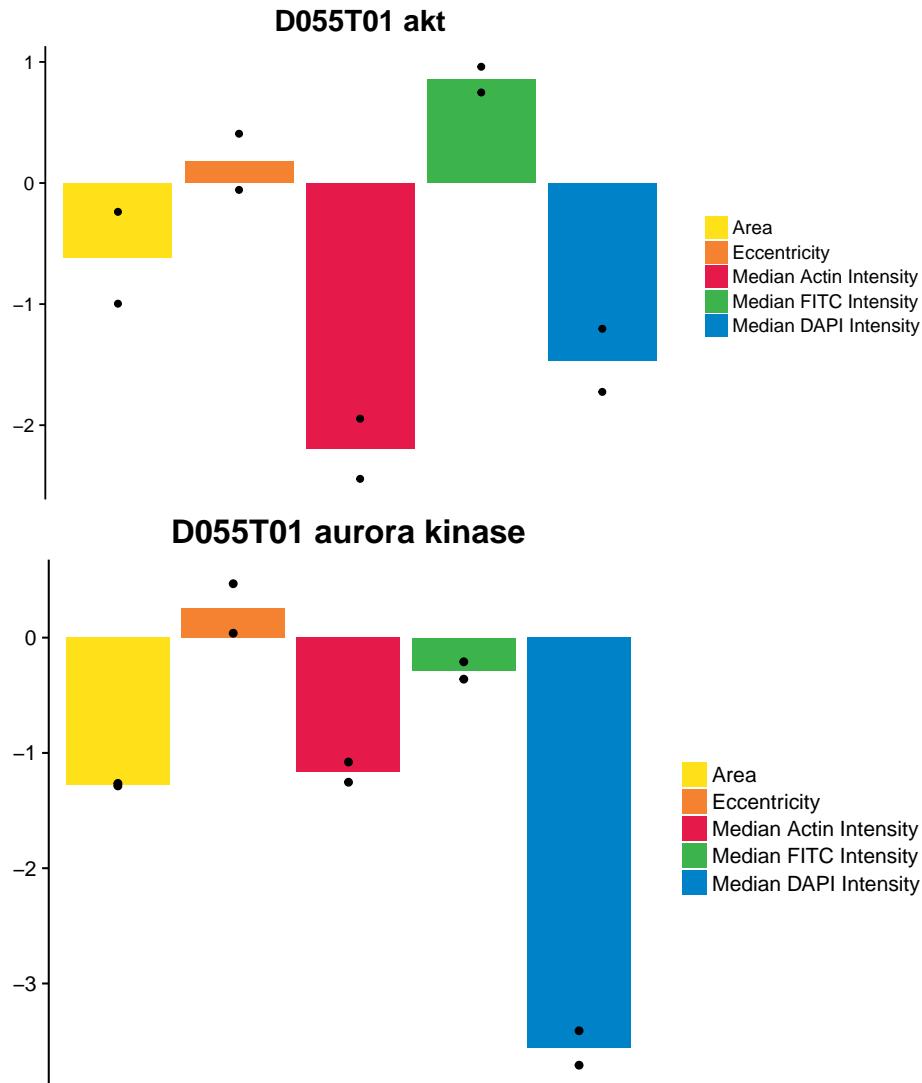
D055T01 gsk-3



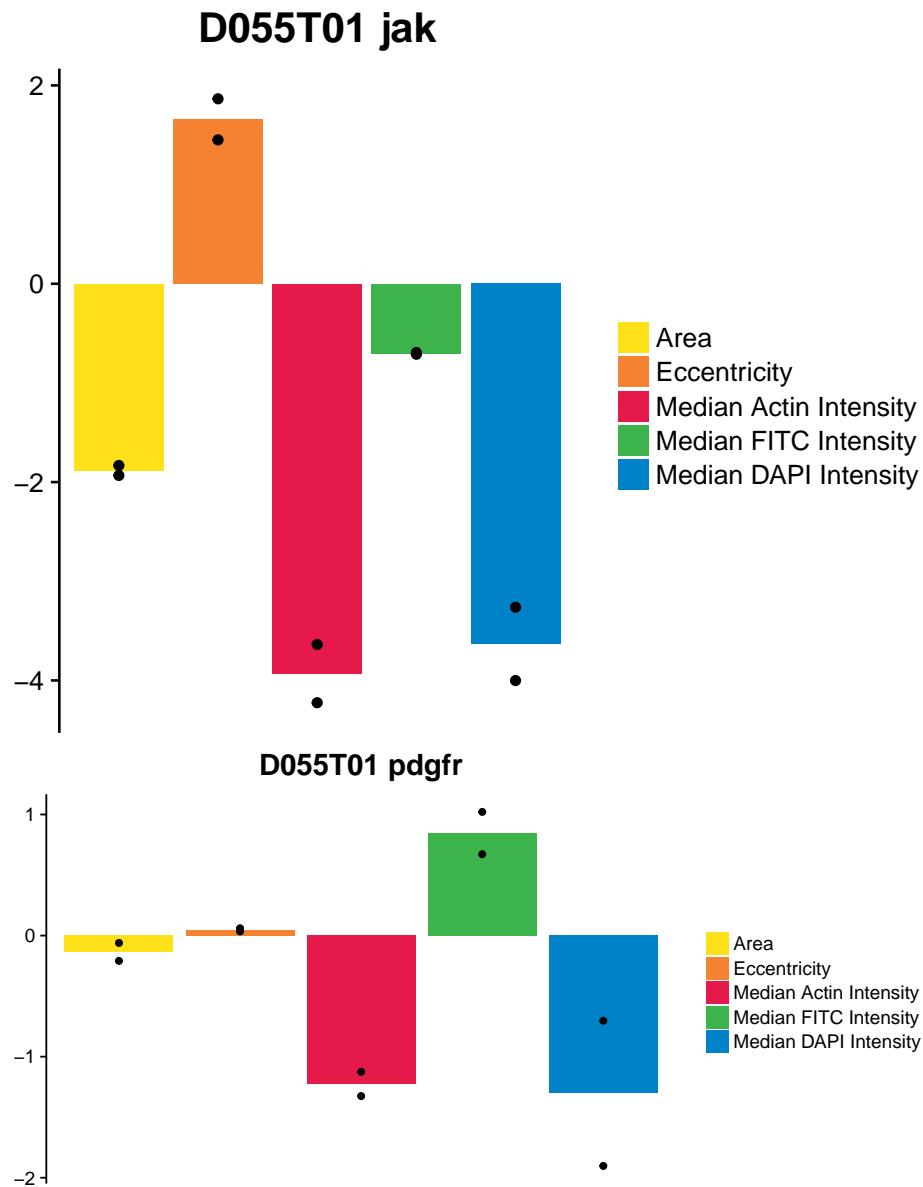
D055T01 cdk



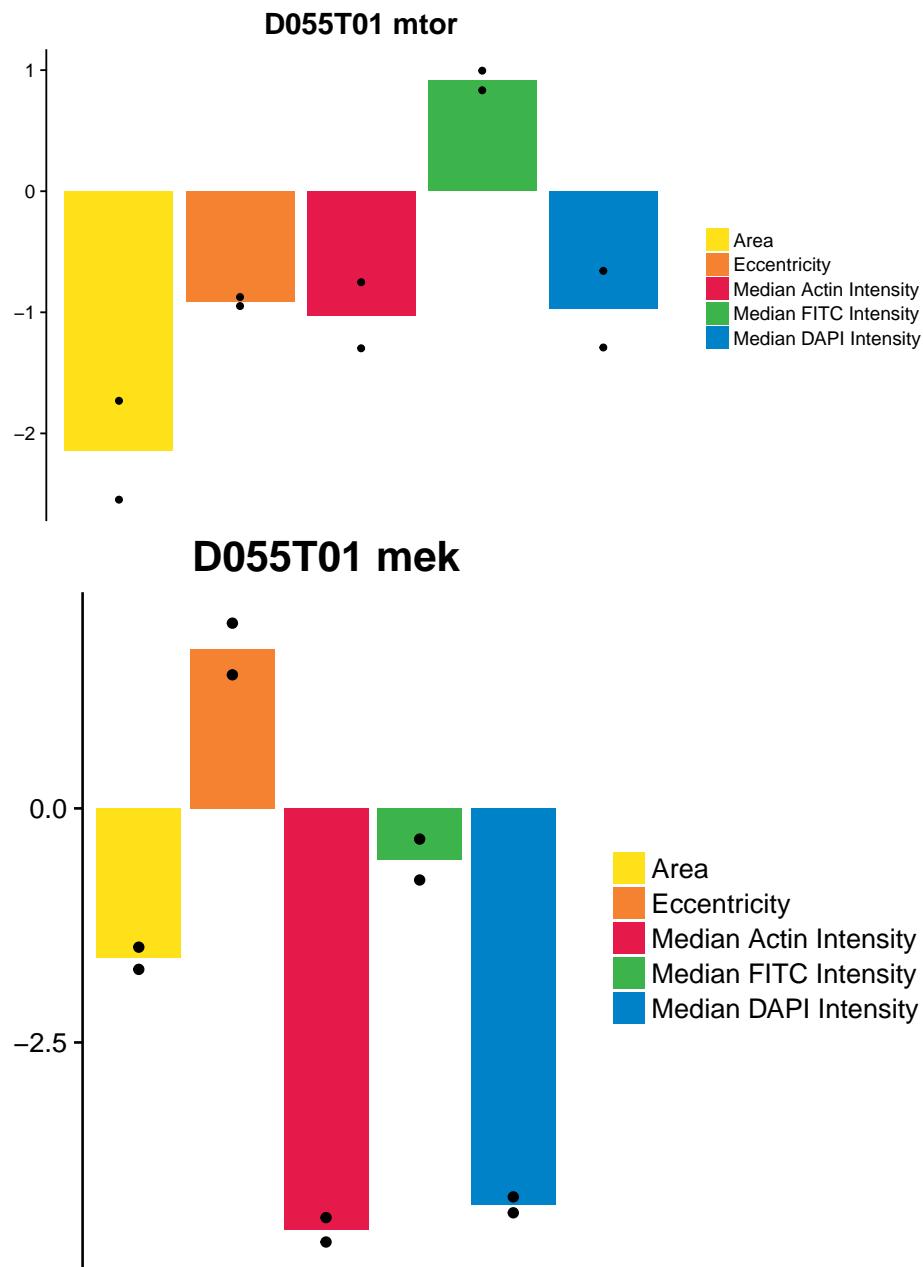
Drug-Induced Phenotypes



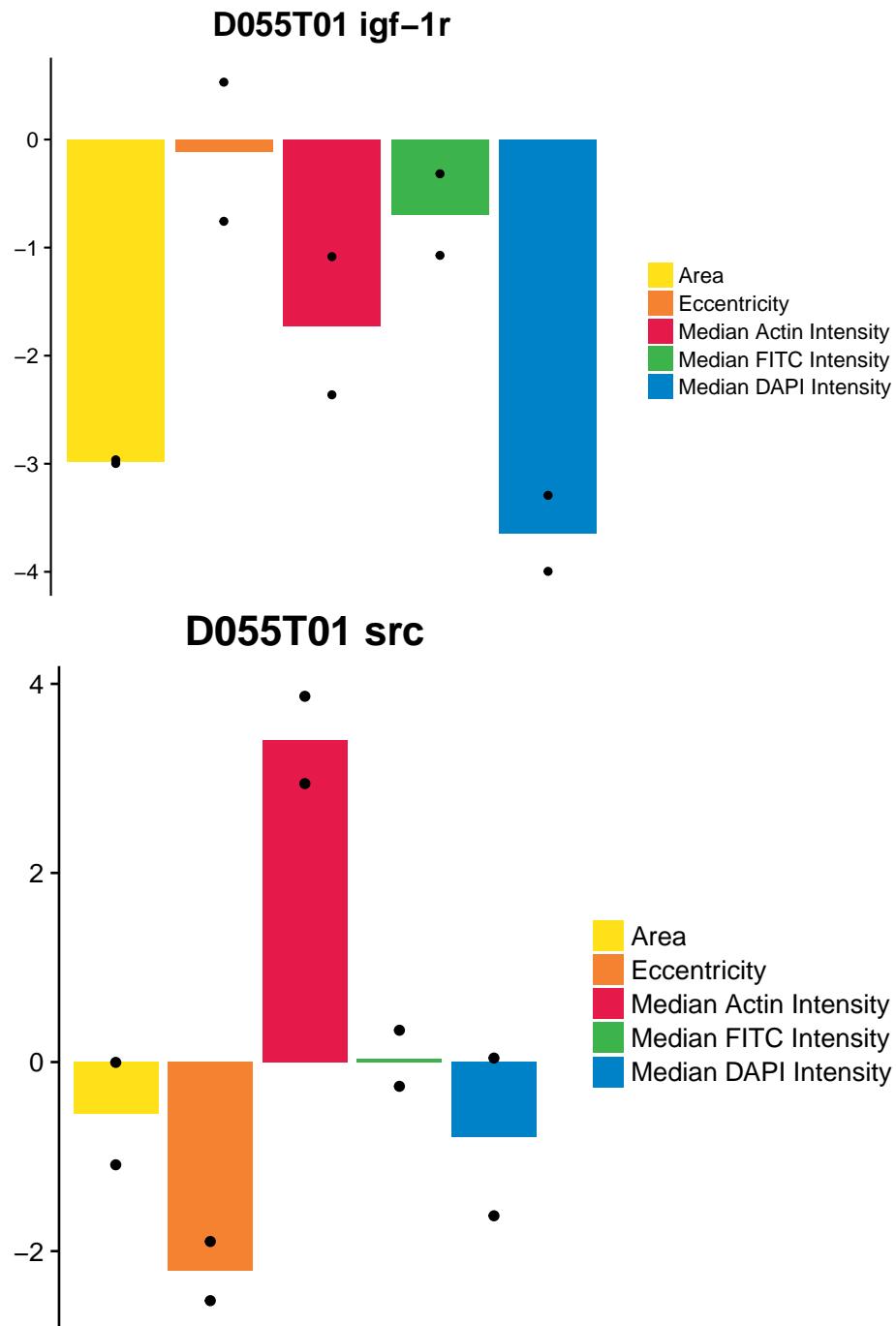
Drug-Induced Phenotypes



Drug-Induced Phenotypes

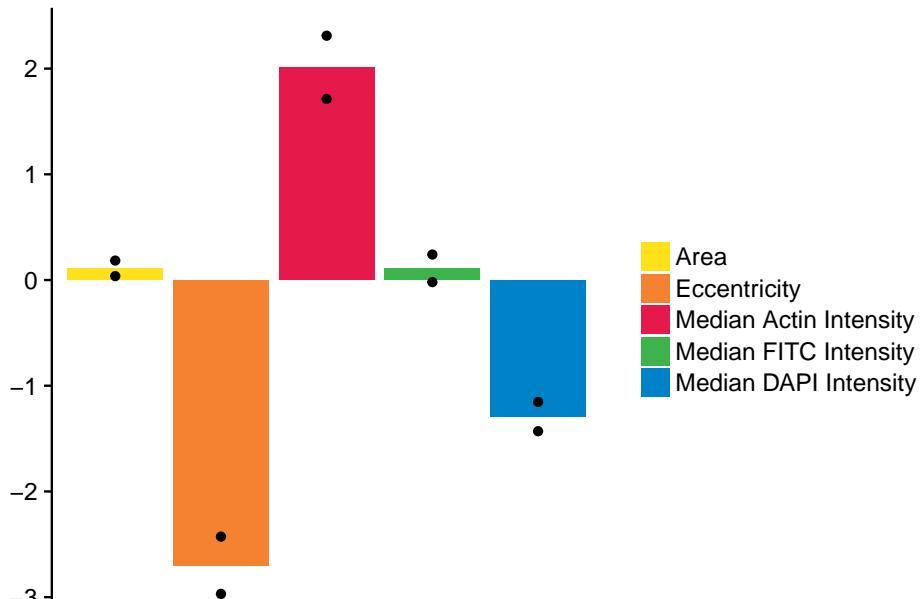


Drug-Induced Phenotypes

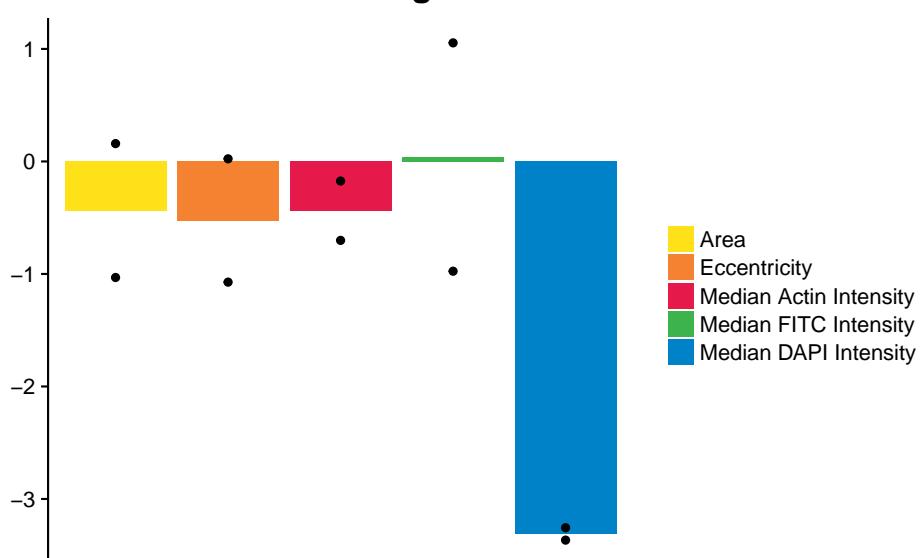


Drug-Induced Phenotypes

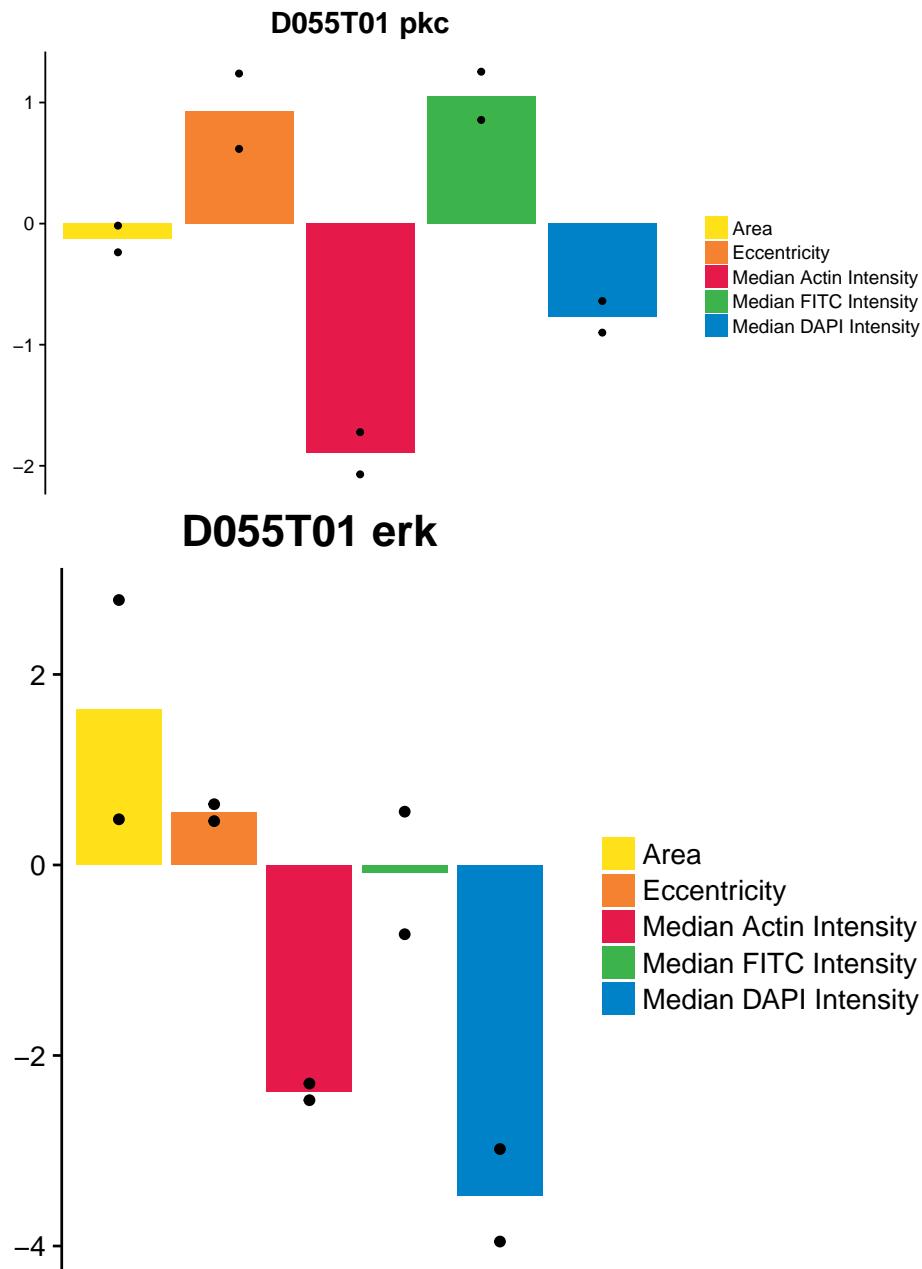
D055T01 bcr-abl



D055T01 vegfr



Drug-Induced Phenotypes



3.4 Drug Effect Differences across all Lines

3.4.1 Comparing Drug Effects Directly

Treating effect vectors of all drugs and all lines equally shows that drug effects on different lines are difficult to compare to each other. Line heterogeneity appears to overpower the effects of drugs. Even the drugs classified as lethal do not cluster together. Notable is that the lines D046T01 and D055T01, which look extremely similar on visual inspection, DO overlap.

Drug-Induced Phenotypes

```
angles = get_angles(profiles_active)
d = acos(angles)*180/pi
hc = hclust(as.dist(d), method = "ward.D2")

cor_pca = cor(profiles_active)
d_pca = as.dist((1 - cor_pca) / 2)
hc_pca = hclust(d_pca, method = "ward.D2")

killing_anno = rep(NA, nrow(metadata_active))
killing_anno[metadata_active$Viability <= 0.2] = "0.2"
killing_anno[
  metadata_active$Viability > 0.2 &
  metadata_active$Viability <= 0.4] = "0.4"
killing_anno[
  metadata_active$Viability > 0.4 &
  metadata_active$Viability <= 0.6] = "0.6"
killing_anno[
  metadata_active$Viability > 0.6 &
  metadata_active$Viability <= 0.8] = "0.8"
killing_anno[
  metadata_active$Viability > 0.8 &
  metadata_active$Viability <= 1] = "1"

annotation = data.frame(
  "Viability" = killing_anno,
  "Pathway" = get_mode_of_action(metadata_active$Drug),
  "Target" = get_targets(metadata_active$Drug),
  "Line" = metadata_active$Line,
  row.names = rownames(profiles_active),
  stringsAsFactors = FALSE)

# Annotate Pathway and Targets
pathways = list()
for(pathway in unique(annotation$Pathway)) {
  pathways[[pathway]] = grep(
    pattern = paste0("\b", pathway, "\b"),
    x = annotation$Pathway,
    ignore.case = TRUE)
}
# My cluster_enrichment function doesn't play well with NA values yet
pathways = pathways[!is.na(names(pathways))]
pathways_sig_vec = get_cluster_enrichment(
  clustering = hc, labels = pathways, min_cluster_size = min_cluster_size)
# 'Others' is too vague
pathways_sig_vec$Labels[pathways_sig_vec$Labels == "Others"] = NA

targets_split = tolower(annotation$Target)
targets = list()
for(target in unique(unlist(strsplit(targets_split, ", ")))) {
  targets[[target]] = grep(
    pattern = paste0("\b", target, "\b"),
```

Drug-Induced Phenotypes

```
x = targets_split, ignore.case = TRUE)
}
# My cluster_enrichment function doesn't play well with NA values yet
targets = targets[!is.na(names(targets))]
targets_sig_vec = get_cluster_enrichment(
  clustering = hc, labels = targets, min_cluster_size = min_cluster_size)

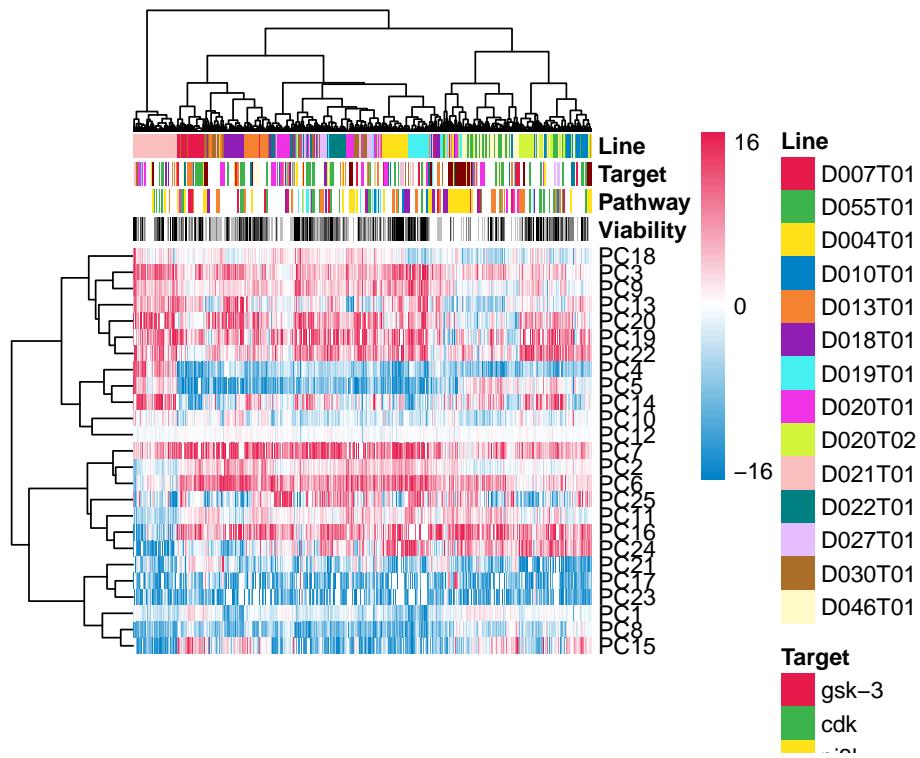
annotation$Pathway = pathways_sig_vec$Labels
annotation$Target = targets_sig_vec$Labels

# Set up colors
anno_colorScale = list(
  "Viability" = setNames(
    object = colorRampPalette(c("black", "white"))(
      length(unique(annotation$Viability))),
    nm = sort(unique(annotation$Viability))))
if(sum(!is.na(annotation$Pathway)) == 0) annotation$Pathway = NULL
if(sum(!is.na(annotation$Target)) == 0) annotation$Target = NULL
for(column in colnames(annotation)) {
  if(column %in% names(anno_colorScale)) next
  entries = na.omit(unique(annotation[[column]]))
  anno_colorScale[[column]] = setNames(
    object = unname(colorScale)[seq_along(entries)],
    nm = entries)
}

# Set up heatmap color scale
sqrt_profiles = profiles_active
sqrt_profiles[sqrt_profiles > 0] = sqrt(sqrt_profiles[sqrt_profiles > 0])
sqrt_profiles[sqrt_profiles < 0] = -sqrt(-sqrt_profiles[sqrt_profiles < 0])
hm_range_limit = max(abs(sqrt_profiles))
hm_breaks = seq(-hm_range_limit, hm_range_limit, length.out = 100)
hm_legend_breaks = c(-floor(hm_range_limit), 0, floor(hm_range_limit))
hm_legend_labels = hm_legend_breaks**2
hm_legend_labels[hm_legend_breaks < 0] = -hm_legend_labels[hm_legend_breaks < 0]
hm_colorscale = colorRampPalette(
  c("#0082c8", "white", "#e6194b"))(length(hm_breaks))

pheatmap(
  mat = t(profiles_active), cluster_rows = hc_pca, cluster_cols = hc,
  show_colnames = FALSE, annotation_col = annotation, breaks = hm_breaks,
  legend_breaks = hm_legend_breaks, legend_labels = hm_legend_labels,
  color = hm_colorscale, annotation_colors = anno_colorScale,
  cellwidth = 0.1, cellheight = 7)
```

Drug-Induced Phenotypes

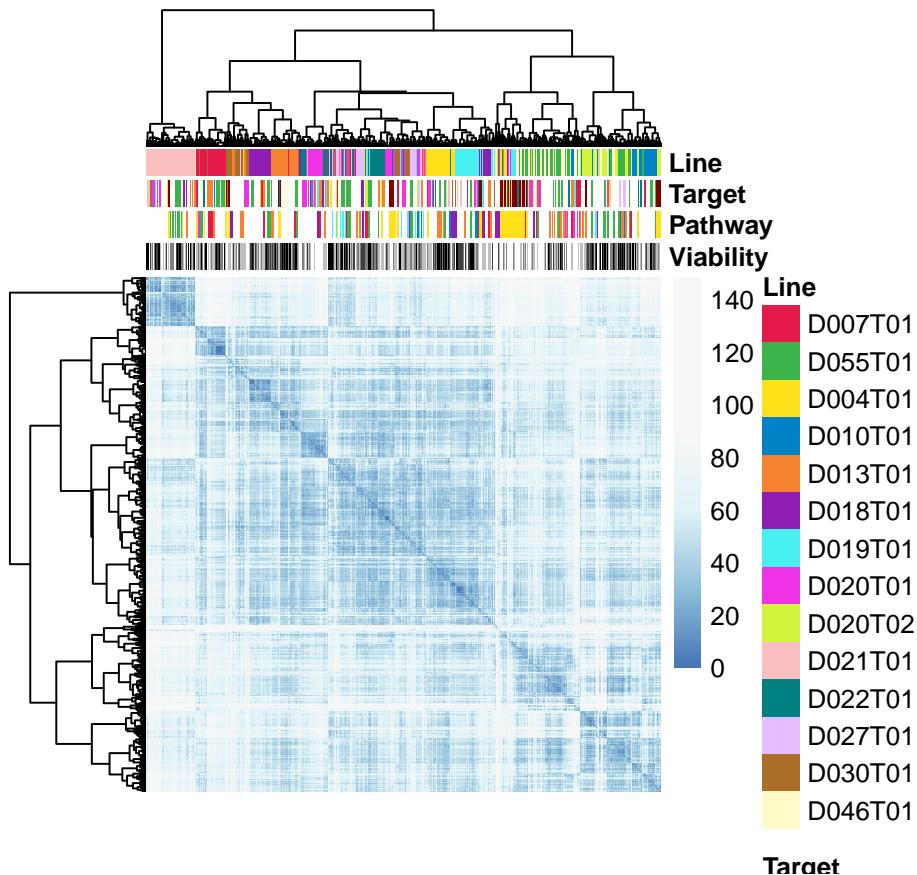


```
if(save_images) pheatmap(
  mat = t(profiles_active), cluster_rows = hc_pca, cluster_cols = hc,
  show_colnames = FALSE, annotation_col = annotation, breaks = hm_breaks,
  legend_breaks = hm_legend_breaks, legend_labels = hm_legend_labels,
  color = hm_colorscale, annotation_colors = anno_colorScale,
  cellwidth = 0.2, cellheight = 7, width = 10, height = 10,
  filename = file.path(
    img_out_dir, "EffectVectorHeatmap_allDrugs_noConcat_PCA.pdf"))

hm_colorscale = colorRampPalette(
  rev(c("#f7f7f7", "#f7f7f7", "#f7f7f7",
        "#E0F3F8", "#91BFDB", "#4575B4")))(150)

pheatmap(
  mat = d, cluster_rows = hc, cluster_cols = hc, color = hm_colorscale,
  show_colnames = FALSE, show_rownames = FALSE,
  annotation_col = annotation, annotation_colors = anno_colorScale,
  cellwidth = 0.1, cellheight = 0.1)
```

Drug-Induced Phenotypes



```
if(save_images) pheatmap(
  mat = d, cluster_rows = hc, cluster_cols = hc, color = hm_colorscale,
  show_colnames = FALSE, show_rownames = FALSE,
  annotation_col = annotation, annotation_colors = anno_colorScale,
  cellwidth = 0.1, cellheight = 0.1, height = 10,
  filename = file.path(
    img_out_dir, "EffectVectorHeatmap_allDrugs_noConcat.pdf"))

```

There is one notably enriched cluster of mTOR inhibitors, in particular the drug “AZD2014” is enriched in 11 cell lines, as well several times in the pan-cell line clustering.

```
sel_indices = metadata_active$Drug == "AZD2014"
drug_profiles = profiles_active[sel_indices, ]
drug_annotation = annotation[sel_indices, ]

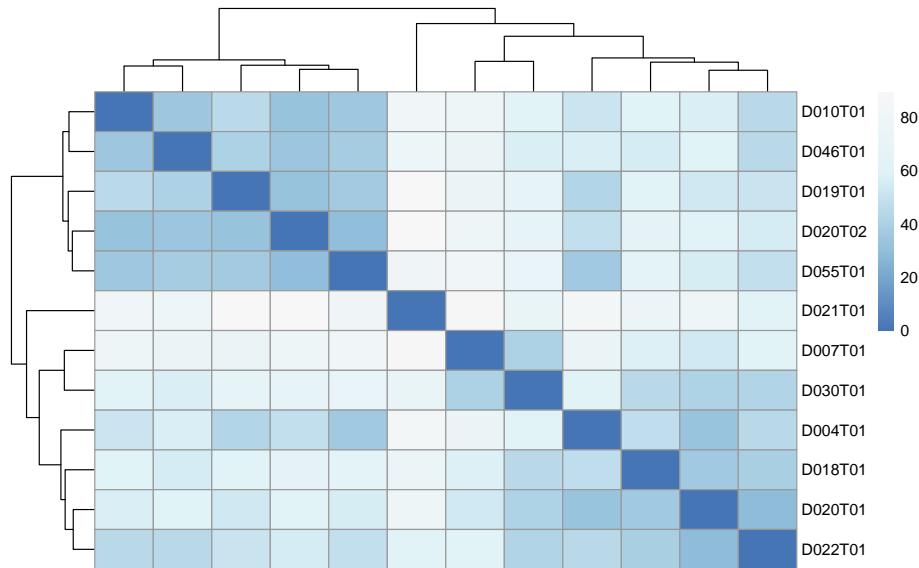
angles_drug = get_angles(drug_profiles)
d_drug = acos(angles_drug) * 180 / pi
hc_drug = hclust(as.dist(d_drug), method = "ward.D2")

hm_colorscale = colorRampPalette(
  rev(c("#f7f7f7", "#E0F3F8", "#91BFDB", "#4575B4")))(150)

pheatmap(
  mat = d_drug, cluster_rows = hc_drug, cluster_cols = hc_drug,
```

Drug-Induced Phenotypes

```
color = hm_colorscale, labels_row = drug_annotation$Line,  
show_colnames = FALSE)
```



3.4.2 Concatenating Vectors

Instead of treating each drug per line separately, I concatenate the effect vectors on all lines for each drug. The resulting composite vector describes the drug's effect on all cell lines without attempting to compare cell lines.

```
profiles_concat = list()  
for(drug in sort(unique(metadata_active$Drug))) {  
# for(drug in sort(unique(metadata$Drug))) {  
  profiles_concat_drug = c()  
  for(line in sort(unique(metadata_active$Line))) {  
# for(line in sort(unique(metadata$Line))) {  
    tmp = profiles[  
      metadata$Drug == drug &  
      metadata$Line == line, , drop = FALSE]  
    if(is.null(nrow(tmp))) stop("Critical Error: the code expects a matrix")  
    if(nrow(tmp) == 0) {  
      tmp = matrix(  
        data = NA, nrow = 1, ncol = n_components,  
        dimnames = list(drug, paste0(line, ".", colnames(tmp))))  
    } else if(nrow(tmp) > 1) {  
      tmp = matrix(  
        colMeans(tmp), nrow=1,  
        dimnames = list(drug, paste0(line, ".", colnames(tmp))))  
    } else {  
      rownames(tmp) = drug  
      colnames(tmp) = paste0(line, ".", colnames(tmp))  
    }  
    profiles_concat_drug = cbind(profiles_concat_drug, as.matrix(tmp))
```

Drug-Induced Phenotypes

```
}

profiles_concat[[drug]] = data.frame(
  profiles_concat_drug)
}
profiles_concat = do.call(rbind, profiles_concat)

# Impute missing values with the means for the corresponding PC of all other
# lines for that drug
missing_drugs = which(is.na(profiles_concat), arr.ind = T)
for(ii in seq_len(nrow(missing_drugs))) {
  entry = missing_drugs[ii, ]
  pc = strsplit(colnames(profiles_concat)[entry[2]], ".", fixed = TRUE)[[1]][2]
  other_pc_cols = sapply(
    strsplit(colnames(profiles_concat), ".", fixed = TRUE),
    "[[", 2) == pc
  mean_val = mean(as.numeric(
    profiles_concat[entry[1], other_pc_cols]), na.rm = TRUE)
  profiles_concat[entry[1], entry[2]] = mean_val
}

# If a drug has no effect in a line, I set its respective PC values to 0
# for(drug in rownames(profiles_concat)) {
#   for(line in sort(unique(metadata_active$Line))) {
#     drug_is_active = metadata$AUC_Mean[
#       metadata$Line == line &
#       metadata$Drug == drug] >= auc_thresh
#     if(length(drug_is_active) == 0) next
#     if(!drug_is_active) profiles_concat[
#       drug, substr(colnames(profiles_concat), 1, 7) == line] = 0
#   }
# }

# Cluster
angles = get_angles(profiles_concat)
d = acos(angles) * 180 / pi
hc = hclust(as.dist(d), method = "ward.D2")

annotation = data.frame(
  "Pathway" = get_mode_of_action(rownames(profiles_concat)),
  "Target" = get_targets(rownames(profiles_concat)),
  row.names = rownames(profiles_concat),
  stringsAsFactors = FALSE)

# Annotate Pathway and Targets
pathways = list()
for(pathway in unique(annotation$Pathway)) {
  pathways[[pathway]] = grep(
    pattern = paste0("\\b", pathway, "\\b"),
    x = annotation$Pathway,
    ignore.case = TRUE)
}
```

Drug-Induced Phenotypes

```
pathways_sig_vec = get_cluster_enrichment(
  clustering = hc, labels = pathways, min_cluster_size = min_cluster_size)
# 'Others' is too vague
pathways_sig_vec$Labels[pathways_sig_vec$Labels == "Others"] = NA

targets_split = tolower(annotation$Target)
targets = list()
for(target in unique(unlist(strsplit(targets_split, ", ")))) {
  targets[[target]] = grep(
    pattern = paste0("\\b", target, "\\b"),
    x = targets_split, ignore.case = TRUE)
}
targets_sig_vec = get_cluster_enrichment(
  clustering = hc, labels = targets, min_cluster_size = min_cluster_size)

annotation$Pathway = pathways_sig_vec$Labels
annotation$Target = targets_sig_vec$Labels

if(sum(!is.na(annotation$Pathway)) == 0) annotation$Pathway = NULL
if(sum(!is.na(annotation$Target)) == 0) annotation$Target = NULL

# Add a summary statistic for lethality
lethality_degree = aggregate(
  x = metadata$Viability <= 0.25,
  by = list("Drug" = metadata$Drug),
  FUN = sum)
rownames(lethality_degree) = lethality_degree$Drug
lethality_degree$Drug = NULL
# Hack to normalize number of lethal lines for positive controls
lethality_degree$x[lethality_degree$x > length(unique(metadata$Line))] = length(unique(metadata$Line))
annotation$Lethality = lethality_degree[rownames(annotation), "x"]

# Set up colors
anno_colorScale = list()
for(column in colnames(annotation)) {
  if(column %in% names(anno_colorScale)) next
  entries = sort(na.omit(unique(annotation[[column]])))
  anno_colorScale[[column]] = setNames(
    object = unname(colorScale)[seq_along(entries)],
    nm = entries)
}
anno_colorScale$Lethality = colorRampPalette(
  c("white", "black"))(length(anno_colorScale$Lethality))

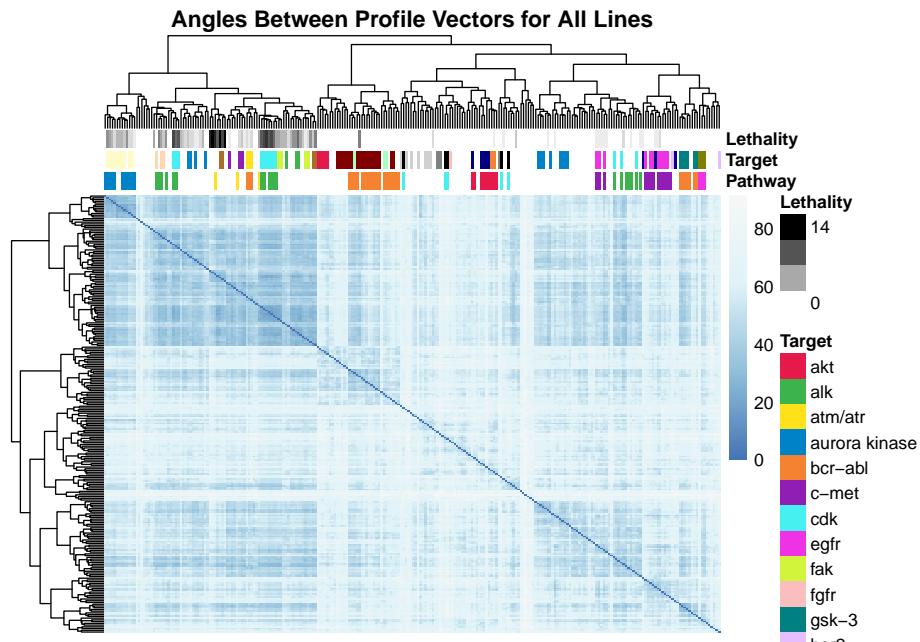
# hm_colorscale = colorRampPalette(
#   rev(c("#f7f7f7", "#f7f7f7", "#f7f7f7", "#f7f7f7",
#         "#E0F3F8", "#91BFDB", "#4575B4")))(150)

hm_colorscale = colorRampPalette(
  rev(c("#f7f7f7", "#E0F3F8", "#91BFDB", "#4575B4")))(150)
```

Drug-Induced Phenotypes

```
anno_colorScale$Target = setNames(  
  object = c("#e6194b", "#3cb44b", "#ffe119", "#0082c8", "#f58231",  
    "#911eb4", "#46f0f0", "#f032e6", "#d2f53c", "#fabebe",  
    "#008080", "#e6beff", "#aa6e28", "#ffffac", "#800000",  
    "#aaffc3", "#808000", "#ffd8b1", "#000080", "#808080",  
    "#d0d0d0", "#000000"),  
  nm = names(anno_colorScale$Target))
```

```
pheatmap(  
  mat = d, annotation_col = annotation,  
  color = hm_colorscale, annotation_colors = anno_colorScale,  
  show_rownames = FALSE, show_colnames = FALSE,  
  cluster_rows = hc, cluster_cols = hc,  
  main = "Angles Between Profile Vectors for All Lines",  
  border_color = NA)
```



```
if(save_images) pheatmap(  
  d, annotation_col = annotation,  
  color = hm_colorscale, annotation_colors = anno_colorScale,  
  show_rownames = FALSE, show_colnames = FALSE,  
  cluster_rows = hc, cluster_cols = hc,  
  main = sprintf("Angles Between Profile Vectors for all lines"),  
  border_color = NA, cellwidth = 1.5, cellheight = 1.5, height = 15,  
  filename = file.path(img_out_dir, "EffectVectorHeatmap_allDrugs.pdf"))
```

Shown are the individual drug viabilities for each line according to the clustering for the drug effects to further showcase the compatibility between the methods

```
active_drugs = rownames(d)  
active_viabilitys = metadata[metadata$Drug %in% active_drugs, ]
```

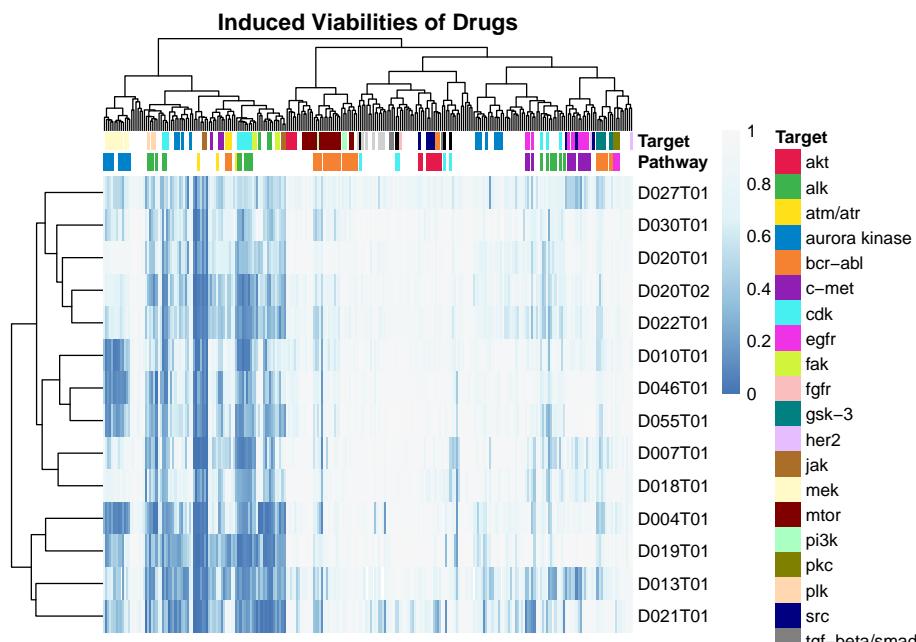
Drug-Induced Phenotypes

```
# Aggregate drugs over wells
active_viabilitys = aggregate(
  x = active_viabilitys$Viability,
  by = list("Drug" = active_viabilitys$Drug,
            "Line" = active_viabilitys$Line),
  FUN = median, na.rm = TRUE)

active_viabilitys = acast(
  data = active_viabilitys,
  formula = Line ~ Drug, value.var = "x")

# Remove viability from annotation
anno_noLethality = annotation
anno_noLethality$Lethality = NULL
anno_colorScale$Lethality = NULL

pheatmap(
  mat = active_viabilitys, annotation_col = anno_noLethality,
  color = hm_colorscale, annotation_colors = anno_colorScale,
  show_rownames = TRUE, show_colnames = FALSE,
  cluster_rows = TRUE, cluster_cols = hc,
  main = "Induced Viabilities of Drugs",
  border_color = NA)
```



```
if(save_images) pheatmap(
  mat = active_viabilitys, annotation_col = anno_noLethality,
  color = hm_colorscale, annotation_colors = anno_colorScale,
  show_rownames = TRUE, show_colnames = FALSE,
  cluster_rows = TRUE, cluster_cols = hc,
```

Drug-Induced Phenotypes

```
main = "Induced Viabilities of Drugs",
border_color = NA, cellwidth = 1.5, cellheight = 10, height = 15,
filename = file.path(img_out_dir, "EffectVectorHeatmap_Viabilities_allDrugs.pdf"))
```

Generate a list of column/row names with annotations

```
annotation_table = annotation[hc$labels[hc$order], ]
colnames(annotation_table) = c(
  "EnrichedPathway", "EnrichedTarget", "Lethality")
annotation_table$Pathway = get_mode_of_action(rownames(annotation_table))
annotation_table$Targets = tolower(get_targets(rownames(annotation_table)))
annotation_table = annotation_table[, c(
  "Pathway", "Targets", "EnrichedPathway", "EnrichedTarget", "Lethality")]
print(annotation_table)
##                                     Pathway
## Ulixertinib (BVD-523, VRT752271)      MAPK
## TAK-733                                MAPK
## Trametinib (GSK1120212)                MAPK
## Cobimetinib (GDC-0973, RG7420)        MAPK
## MEK162 (ARRY-162, ARRY-438162)        MAPK
## PD0325901                               DNA Damage
## Refametinib (RDEA119, Bay 86-9766)    Others
## AZD8330                                 MAPK
## Pimasertib (AS-703026)                 MAPK
## RAF265 (CHIR-265)                      MAPK
## Selumetinib (AZD6244)                  MAPK
## PD318088                                 MAPK
## VX-11e                                   MAPK
## NVP-BHG712                               Protein Tyrosine Kinase
## YM201636                                 PI3K/Akt/mTOR
## Lapatinib (GW-572016) Ditosylate       Protein Tyrosine Kinase
## PF-573228                               Angiogenesis
## BMS-265246                               Cell Cycle
## KU-60019                                 DNA Damage
## Triciribine                             Others
## CUDC-101                                Epigenetics
## GSK461364                               Cell Cycle
## AZD7762                                 Cell Cycle
## Rigosertib (ON-01910)                   Cell Cycle
## BI 2536                                  Others
## JNJ-7706621                            Cell Cycle
## PIK-93                                  PI3K/Akt/mTOR
## PF-4708671                            PI3K/Akt/mTOR
## Flavopiridol (Alvocidib)               Cell Cycle
## AT7519                                  Cell Cycle
## SNS-032 (BMS-387032)                   Others
## Staurosporine_500nM                     <NA>
## BMS-754807                               Others
## LY3023414                               PI3K/Akt/mTOR
## Hesperadin                             Cell Cycle
## KW-2449                                 Angiogenesis
```

Drug-Induced Phenotypes

## Crizotinib (PF-02341066)	Others
## ENMD-2076	Angiogenesis
## HMN-214	Cell Cycle
## BKM120 (NVP-BKM120, Buparlisib)	PI3K/Akt/mTOR
## NVP-AEW541	Protein Tyrosine Kinase
## Alisertib (MLN8237)	Others
## GSK2292767	PI3K/Akt/mTOR
## BGT226 (NVP-BGT226)	PI3K/Akt/mTOR
## Bortezomib	<NA>
## CUDC-907	Cytoskeletal Signaling
## CHIR-124	Cell Cycle
## Pacritinib (SB1518)	JAK/STAT
## AT9283	Others
## Irinotecan / SN-38	<NA>
## JNK Inhibitor IX	MAPK
## Tivantinib (ARQ 197)	Protein Tyrosine Kinase
## HS-173	PI3K/Akt/mTOR
## Wnt agonist 1	Stem Cells & Wnt
## FRAX597	Cytoskeletal Signaling
## Tepotinib (EMD 1214063)	Protein Tyrosine Kinase
## Golvatinib (E7050)	Protein Tyrosine Kinase
## KX2-391	Angiogenesis
## VE-822	PI3K/Akt/mTOR
## AZ20	PI3K/Akt/mTOR
## AZD6738	PI3K/Akt/mTOR
## AMG-900	Cell Cycle
## PF-543	GPCR & G Protein
## PF-3758309	Cytoskeletal Signaling
## AZD5438	Cell Cycle
## R547	Cell Cycle
## A-674563	PI3K/Akt/mTOR
## Dinaciclib (SCH727965)	Cell Cycle
## P276-00	Cell Cycle
## Flavopiridol HCl	Cell Cycle
## PHA-767491	Cell Cycle
## PF-00562271	Angiogenesis
## PF-562271	Angiogenesis
## IKK-16 (IKK Inhibitor VII)	NF- <u>B</u>
## LDK378	Protein Tyrosine Kinase
## Vacquinol-1	MAPK
## GSK2126458 (GSK458)	PI3K/Akt/mTOR
## Torin 2	PI3K/Akt/mTOR
## AZD3463	Protein Tyrosine Kinase
## AP26113	Protein Tyrosine Kinase
## SC1	MAPK
## SGI-7079	Protein Tyrosine Kinase
## PF-431396	Angiogenesis
## TAE226 (NVP-TAE226)	Angiogenesis
## Milciclib (PHA-848125)	Cell Cycle
## TG101209	JAK/STAT
## TG101348 (SAR302503)	JAK/STAT

Drug-Induced Phenotypes

```

## GSK690693          Others
## GDC-0068          PI3K/Akt/mTOR
## MK-2206 2HCl       Others
## AZD5363          PI3K/Akt/mTOR
## Uprosertib (GSK2141795) PI3K/Akt/mTOR
## BMS-536924         Protein Tyrosine Kinase
## OSI-906 (Linsitinib) Others
## ZSTK474           Neuronal Signaling
## KU-0063794        PI3K/Akt/mTOR
## WYE-354            PI3K/Akt/mTOR
## Temsirolimus (CCI-779, NSC 683864) Neuronal Signaling
## Everolimus (RAD001) Others
## Ridaforolimus (Deforolimus, MK-8669) PI3K/Akt/mTOR
## GDC-0980 (RG7422) PI3K/Akt/mTOR
## INK 128 (MLN0128) PI3K/Akt/mTOR
## VS-5584 (SB2343)  PI3K/Akt/mTOR
## PF-04691502        PI3K/Akt/mTOR
## PP121              Protein Tyrosine Kinase
## CC-223             PI3K/Akt/mTOR
## PP242              PI3K/Akt/mTOR
## AZD2014            PI3K/Akt/mTOR
## GDC-0349            PI3K/Akt/mTOR
## OSI-027             PI3K/Akt/mTOR
## AZD8055            PI3K/Akt/mTOR
## CH5132799          PI3K/Akt/mTOR
## WYE-125132 (WYE-132) PI3K/Akt/mTOR
## CO-1686 (AVL-301)  Protein Tyrosine Kinase
## A66                PI3K/Akt/mTOR
## BYL719              PI3K/Akt/mTOR
## Akti-1/2            PI3K/Akt/mTOR
## WAY-600             PI3K/Akt/mTOR
## ETP-46464          PI3K/Akt/mTOR
## CCT128930          PI3K/Akt/mTOR
## GSK2334470          PI3K/Akt/mTOR
## Nintedanib (BIBF 1120)_uncertain Protein Tyrosine Kinase
## WIKI4               Stem Cells & Wnt
## Axitinib            Protein Tyrosine Kinase
## PHA-665752          Others
## PD173074            Angiogenesis
## AT13148             PI3K/Akt/mTOR
## AT7867              PI3K/Akt/mTOR
## Linifanib (ABT-869) Protein Tyrosine Kinase
## PD184352 (CI-1040) MAPK
## U0126-EtOH          Others
## Dovitinib (TKI-258, CHIR-258) Angiogenesis
## Sunitinib Malate     Microbiology
## Cediranib (AZD2171) Protein Tyrosine Kinase
## SGI-1776 free base JAK/STAT
## GSK2636771          PI3K/Akt/mTOR
## RepSox              TGF-beta/Smad
## K02288              TGF-beta/Smad

```

Drug-Induced Phenotypes

## TWS119	PI3K/Akt/mTOR
## ICG-001	Stem Cells & Wnt
## PRI-724	Stem Cells & Wnt
## AZD4547	Angiogenesis
## CEP-33779	JAK/STAT
## YO-01027	Proteases
## WZ4003	PI3K/Akt/mTOR
## AVL-292	Angiogenesis
## ASP3026	Protein Tyrosine Kinase
## Entrectinib (RXDX-101)	Protein Tyrosine Kinase
## BIO	PI3K/Akt/mTOR
## Semaxanib (SU5416)	Protein Tyrosine Kinase
## SU6656	Angiogenesis
## DCC-2036 (Rebastinib)	Angiogenesis
## Purvalanol A	Cell Cycle
## Dabrafenib (GSK2118436)	MAPK
## PP1	Angiogenesis
## PP2	Angiogenesis
## WH-4-023	Angiogenesis
## Dasatinib	Angiogenesis
## PD173955	Angiogenesis
## Ponatinib (AP24534)	Angiogenesis
## PRT062607 (P505-15, BIIB057) HCl	Angiogenesis
## LDN-214117	TGF-beta/Smad
## LGK-974	Stem Cells & Wnt
## AZ 960	JAK/STAT
## DASA-58	Others
## FH535	Stem Cells & Wnt
## Mubritinib (TAK 165)	Protein Tyrosine Kinase
## VPS34-IN1	PI3K/Akt/mTOR
## IMD 0354	NF- <u>_</u> B
## Tyrphostin 9	Protein Tyrosine Kinase
## Pazopanib	Protein Tyrosine Kinase
## Sorafenib	MAPK
## SU11274	Neuronal Signaling
## AG-1024	Protein Tyrosine Kinase
## Amuvatinib (MP-470)	Protein Tyrosine Kinase
## Sorafenib Tosylate	Neuronal Signaling
## Foretinib (GSK1363089)	Others
## SNS-314 Mesylate	Others
## CYC116	Cell Cycle
## VX-680 (Tozastertib, MK-0457)	Endocrinology & Hormones
## BX-912	PI3K/Akt/mTOR
## AZD1480	JAK/STAT
## Danusertib (PHA-739358)	Others
## Quizartinib (AC220)	Angiogenesis
## AEE788 (NVP-AEE788)	Protein Tyrosine Kinase
## WZ3146	Protein Tyrosine Kinase
## ZM 447439	Others
## Barasertib (AZD1152-HQPA)	Others
## PHA-680632	Cell Cycle

Drug-Induced Phenotypes

## Aurora A Inhibitor I	Cell Cycle
## OSU-03012 (AR-12)	Others
## MK-5108 (VX-689)	Cell Cycle
## HTH-01-015	PI3K/Akt/mTOR
## MK-8745	Cell Cycle
## TPCA-1	NF- <u>_</u> B
## BMS-345541	NF- <u>_</u> B
## Dovitinib (TKI-258) Dilactic Acid	Angiogenesis
## GSK1838705A	Protein Tyrosine Kinase
## NVP-BSK805 2HCl	JAK/STAT
## LY2784544	JAK/STAT
## SU9516	Cell Cycle
## CNX-2006	Protein Tyrosine Kinase
## Dacomitinib (PF299804, PF299)	Protein Tyrosine Kinase
## JNK-IN-8	MAPK
## AZD9291	Protein Tyrosine Kinase
## GZD824	Angiogenesis
## ERK5-IN-1	MAPK
## XMD8-92	MAPK
## LY2835219	Cell Cycle
## CP-673451	Protein Tyrosine Kinase
## Crenolanib (CP-868596)	Protein Tyrosine Kinase
## PHA-793887	Cell Cycle
## Ro 31-8220 Mesylate	TGF-beta/Smad
## Ro3280	Cell Cycle
## LY2603618	Cell Cycle
## PF-477736	Cell Cycle
## TAK-632	MAPK
## LDC000067	Cell Cycle
## Fostamatinib (R788)	Angiogenesis
## TAK-901	Cell Cycle
## Bosutinib (SKI-606)	Angiogenesis
## AST-1306	Protein Tyrosine Kinase
## Imatinib Mesylate (STI571)	Protein Tyrosine Kinase
## AZD8931 (Sapitinib)	Protein Tyrosine Kinase
## Gefitinib (ZD1839)	Protein Tyrosine Kinase
## Saracatinib (AZD0530)	Angiogenesis
## OSI-420	Protein Tyrosine Kinase
## Pelitinib (EKB-569)	Protein Tyrosine Kinase
## AZD3759	Protein Tyrosine Kinase
## PD168393	Protein Tyrosine Kinase
## AG-1478 (Tyrphostin AG-1478)	Protein Tyrosine Kinase
## ZM 306416	Protein Tyrosine Kinase
## AC480 (BMS-599626)	Neuronal Signaling
## Ibrutinib (PCI-32765)	Angiogenesis
## ONO-4059	Angiogenesis
## LY2090314	PI3K/Akt/mTOR
## AZD2858	PI3K/Akt/mTOR
## CHIR-98014	PI3K/Akt/mTOR
## AZD1080	PI3K/Akt/mTOR
## CHIR-99021 (CT99021) HCl	PI3K/Akt/mTOR

Drug-Induced Phenotypes

## Enzastaurin (LY317615)	Neuronal Signaling
## 1-Azakenpaulone	PI3K/Akt/mTOR
## CHIR-99021 (CT99021)	PI3K/Akt/mTOR
## Sotrastaurin	TGF-beta/Smad
## GF109203X	TGF-beta/Smad
## Go 6983	TGF-beta/Smad
## AZ 628	MAPK
## MLN2480	MAPK
## CX-6258 HCl	JAK/STAT
## Piceatannol	Angiogenesis
## PQ 401	Protein Tyrosine Kinase
## Tyrphostin AG 879	Protein Tyrosine Kinase
##	
## Ulixertinib (BVD-523, VRT752271)	Targets
## TAK-733	erk
## Trametinib (GSK1120212)	mek
## Cobimetinib (GDC-0973, RG7420)	mek
## MEK162 (ARRY-162, ARRY-438162)	mek
## PD0325901	mek
## Refametinib (RDEA119, Bay 86-9766)	mek
## AZD8330	mek
## Pimasertib (AS-703026)	mek
## RAF265 (CHIR-265)	raf, vegfr
## Selumetinib (AZD6244)	mek
## PD318088	mek
## VX-11e	mek
## NVP-BHG712	vegfr, src, raf, bcr-abl
## YM201636	pi3k
## Lapatinib (GW-572016) Ditosylate	egfr, her2
## PF-573228	fak
## BMS-265246	cdk
## KU-60019	atm
## Triciribine	akt
## CUDC-101	hdac, egfr, her2
## GSK461364	plk
## AZD7762	chk
## Rigosertib (ON-01910)	plk
## BI 2536	plk
## JNJ-7706621	cdk, aurora kinase
## PIK-93	pi3k, vegfr
## PF-4708671	s6 kinase
## Flavopiridol (Alvocidib)	cdk
## AT7519	cdk
## SNS-032 (BMS-387032)	cdk
## Staurosporine_500nM	<NA>
## BMS-754807	igf-1r
## LY3023414	akt
## Hesperadin	aurora kinase
## KW-2449	flt, bcr-abl, aurora kinase
## Crizotinib (PF-02341066)	c-met, alk
## ENMD-2076	flt, aurora kinase, vegfr

Drug-Induced Phenotypes

```
## HMN-214 plk
## BKM120 (NVP-BKM120, Buparlisib) pi3k
## NVP-AEW541 igf-1r
## Alisertib (MLN8237) aurora kinase
## GSK2292767 pi3k
## BGT226 (NVP-BGT226) pi3k, mtor
## Bortezomib psmb1
## CUDC-907 hdac, pi3k
## CHIR-124 chk
## Pacritinib (SB1518) jak
## AT9283 bcr-abl, jak, aurora kinase
## Irinotecan / SN-38 top1mt
## JNK Inhibitor IX jnk
## Tivantinib (ARQ 197) c-met
## HS-173 pi3k
## Wnt agonist 1 wnt/beta-catenin
## FRAX597 pak
## Tepotinib (EMD 1214063) c-met
## Golvatinib (E7050) c-met, vegfr
## KX2-391 src
## VE-822 atm/atr
## AZ20 atm/atr
## AZD6738 atm/atr
## AMG-900 aurora kinase
## PF-543 sphk1
## PF-3758309 pak
## AZD5438 cdk
## R547 cdk
## A-674563 akt, cdk, pka
## Dinaciclib (SCH727965) cdk
## P276-00 cdk
## Flavopiridol HCl cdk
## PHA-767491 cdk
## PF-00562271 fak
## PF-562271 fak
## IKK-16 (IKK Inhibitor VII) ikk
## LDK378 alk
## Vacquinol-1 jnk
## GSK2126458 (GSK458) pi3k, mtor
## Torin 2 mtor
## AZD3463 alk
## AP26113 alk
## SC1 erk
## SGI-7079 vegfr
## PF-431396 fak
## TAE226 (NVP-TAE226) fak
## Milciclib (PHA-848125) cdk
## TG101209 flt, jak, c-ret
## TG101348 (SAR302503) jak
## GSK690693 akt
## GDC-0068 akt
```

Drug-Induced Phenotypes

```
## MK-2206 2HCl                      akt
## AZD5363                           akt
## Uprosertib (GSK2141795)            akt
## BMS-536924                         igf-1r
## OSI-906 (Linsitinib)              igf-1r
## ZSTK474                           pi3k
## KU-0063794                         mtor
## WYE-354                            mtor
## Temsirolimus (CCI-779, NSC 683864) mtor
## Everolimus (RAD001)                mtor
## Ridaforolimus (Deforolimus, MK-8669) mtor
## GDC-0980 (RG7422)                  mtor, pi3k
## INK 128 (MLN0128)                  mtor
## VS-5584 (SB2343)                  pi3k
## PF-04691502                       mtor, pi3k, akt
## PP121                             dna-pk, mtor, pdgf
## CC-223                           mtor
## PP242                            mtor
## AZD2014                           mtor
## GDC-0349                           mtor
## OSI-027                            mtor
## AZD8055                           mtor
## CH5132799                         pi3k, mtor
## WYE-125132 (WYE-132)              mtor
## CO-1686 (AVL-301)                 egfr
## A66                                pi3k
## BYL719                            pi3k
## Akti-1/2                           akt
## WAY-600                            mtor
## ETP-46464                          mtor
## CCT128930                          akt
## GSK2334470                         pdk-1
## Nintedanib (BIBF 1120)_uncertain vegfr, pdgfr, fgfr
## WIKI4                            wnt/beta-catenin
## Axitinib                          vegfr, pdgfr, c-kit
## PHA-665752                         c-met
## PD173074                          fgfr, vegfr
## AT13148                           akt
## AT7867                            akt, s6 kinase
## Linifanib (ABT-869)               pdgfr, vegfr
## PD184352 (CI-1040)                mek
## U0126-EtOH                         mek
## Dovitinib (TKI-258, CHIR-258)     c-kit, fgfr, flt, vegfr, pdgfr
## Sunitinib Malate                   vegfr, pdgfr, c-kit, flt
## Cediranib (AZD2171)               vegfr, flt
## SGI-1776 free base                pim
## GSK2636771                         pi3k
## RepSox                            tgf-beta/smad
## K02288                            tgf-beta/smad
## TWS119                            gsk-3
## ICG-001                           wnt/beta-catenin
```

Drug-Induced Phenotypes

```

## PRI-724 wnt/beta-catenin
## AZD4547 fgfr
## CEP-33779 jak
## YO-01027 gamma-secretase
## WZ4003 ampk
## AVL-292 btk
## ASP3026 alk
## Entrectinib (RXDX-101) trk receptor
## BIO gsk-3
## Semaxanib (SU5416) vegfr
## SU6656 src
## DCC-2036 (Rebastinib) bcr-abl
## Purvalanol A cdk
## Dabrafenib (GSK2118436) raf
## PP1 src
## PP2 src
## WH-4-023 src
## Dasatinib src, bcr-abl, c-kit
## PD173955 bcr-abl
## Ponatinib (AP24534) bcr-abl, vegfr, fgfr, pdgfr, flt
## PRT062607 (P505-15, BIIB057) HCl syk
## LDN-214117 tgf-beta/smad
## LGK-974 wnt/beta-catenin
## AZ 960 jak
## DASA-58 pkm2
## FH535 wnt/beta-catenin
## Mubritinib (TAK 165) her2
## VPS34-IN1 pi3k
## IMD 0354 ikk
## Tyrphostin 9 egfr
## Pazopanib vegfr
## Sorafenib raf
## SU11274 c-met
## AG-1024 igf-1r
## Amuvatinib (MP-470) c-met, c-kit, pdgfr, flt, c-ret
## Sorafenib Tosylate vegfr, pdgfr, raf
## Foretinib (GSK1363089) c-met, vegfr
## SNS-314 Mesylate aurora kinase
## CYC116 aurora kinase, vegfr
## VX-680 (Tozasertib, MK-0457) aurora kinase
## BX-912 pdk-1
## AZD1480 jak
## Danusertib (PHA-739358) aurora kinase, fgfr, bcr-abl, c-ret, src
## Quizartinib (AC220) flt
## AEE788 (NVP-AEE788) egfr, flt, vegfr, her2
## WZ3146 egfr
## ZM 447439 aurora kinase
## Barasertib (AZD1152-HQPA) aurora kinase
## PHA-680632 aurora kinase
## Aurora A Inhibitor I aurora kinase
## OSU-03012 (AR-12) pdk-1

```

Drug-Induced Phenotypes

## MK-5108 (VX-689)	aurora kinase
## HTH-01-015	ampk
## MK-8745	aurora kinase
## TPCA-1	ikk
## BMS-345541	i_b/ikk
## Dovitinib (TKI-258) Dilactic Acid	flt, fgfr, pdgfr, vegfr, c-kit
## GSK1838705A	igf-1, alk
## NVP-BSK805 2HCl	jak
## LY2784544	jak
## SU9516	cdk
## CNX-2006	egfr
## Dacomitinib (PF299804, PF299)	egfr
## JNK-IN-8	jnk
## AZD9291	egfr
## GZD824	bcr-abl
## ERK5-IN-1	erk
## XMD8-92	erk
## LY2835219	cdk
## CP-673451	pdgfr
## Crenolanib (CP-868596)	pdgfr
## PHA-793887	cdk
## Ro 31-8220 Mesylate	pkc
## Ro3280	plk
## LY2603618	chk
## PF-477736	chk
## TAK-632	raf
## LDC000067	cdk
## Fostamatinib (R788)	syk
## TAK-901	aurora kinase
## Bosutinib (SKI-606)	src
## AST-1306	egfr
## Imatinib Mesylate (STI571)	pdgfr, c-kit, bcr-abl
## AZD8931 (Sapitinib)	egfr, her2
## Gefitinib (ZD1839)	egfr
## Saracatinib (AZD0530)	src, bcr-abl
## OSI-420	egfr
## Pelitinib (EKB-569)	egfr
## AZD3759	egfr
## PD168393	egfr
## AG-1478 (Tyrphostin AG-1478)	egfr
## ZM 306416	vegfr
## AC480 (BMS-599626)	her2
## Ibrutinib (PCI-32765)	src
## ONO-4059	btk
## LY2090314	gsk-3
## AZD2858	gsk-3
## CHIR-98014	gsk-3
## AZD1080	gsk-3
## CHIR-99021 (CT99021) HCl	pi3k/akt/mTOR
## Enzastaurin (LY317615)	pkc
## 1-Azakenpaullone	gsk-3

Drug-Induced Phenotypes

## CHIR-99021 (CT99021)		gsk-3
## Sotрастaurин		pkc
## GF109203X		pkc
## Go 6983		pkc
## AZ 628		raf
## MLN2480		raf
## CX-6258 HCl		pim
## Piceatannol		syk
## PQ 401		igf-1r
## Tyrphostin AG 879		her2
##	EnrichedPathway	
## Ulixertinib (BVD-523, VRT752271)	MAPK	
## TAK-733	MAPK	
## Trametinib (GSK1120212)	MAPK	
## Cobimetinib (GDC-0973, RG7420)	MAPK	
## MEK162 (ARRY-162, ARRY-438162)	MAPK	
## PD0325901	<NA>	
## Refametinib (RDEA119, Bay 86-9766)	<NA>	
## AZD8330	MAPK	
## Pimasertib (AS-703026)	MAPK	
## RAF265 (CHIR-265)	MAPK	
## Selumetinib (AZD6244)	MAPK	
## PD318088	MAPK	
## VX-11e	MAPK	
## NVP-BHG712	<NA>	
## YM201636	<NA>	
## Lapatinib (GW-572016) Ditosylate	<NA>	
## PF-573228	<NA>	
## BMS-265246	<NA>	
## KU-60019	<NA>	
## Triciribine	<NA>	
## CUDC-101	<NA>	
## GSK461364	Cell Cycle	
## AZD7762	Cell Cycle	
## Rigosertib (ON-01910)	Cell Cycle	
## BI 2536	<NA>	
## JNJ-7706621	Cell Cycle	
## PIK-93	<NA>	
## PF-4708671	<NA>	
## Flavopiridol (Alvocidib)	Cell Cycle	
## AT7519	Cell Cycle	
## SNS-032 (BMS-387032)	<NA>	
## Staurosporine_500nM	<NA>	
## BMS-754807	<NA>	
## LY3023414	<NA>	
## Hesperadin	<NA>	
## KW-2449	<NA>	
## Crizotinib (PF-02341066)	<NA>	
## ENMD-2076	<NA>	
## HMN-214	<NA>	
## BKM120 (NVP-BKM120, Buparlisib)	<NA>	

Drug-Induced Phenotypes

## NVP-AEW541	<NA>
## Alisertib (MLN8237)	<NA>
## GSK2292767	<NA>
## BGT226 (NVP-BGT226)	<NA>
## Bortezomib	<NA>
## CUDC-907	Cytoskeletal Signaling
## CHIR-124	<NA>
## Pacritinib (SB1518)	<NA>
## AT9283	<NA>
## Irinotecan / SN-38	<NA>
## JNK Inhibitor IX	<NA>
## Tivantinib (ARQ 197)	<NA>
## HS-173	<NA>
## Wnt agonist 1	<NA>
## FRAX597	Cytoskeletal Signaling
## Tepotinib (EMD 1214063)	<NA>
## Golvatinib (E7050)	<NA>
## KX2-391	<NA>
## VE-822	PI3K/Akt/mTOR
## AZ20	PI3K/Akt/mTOR
## AZD6738	PI3K/Akt/mTOR
## AMG-900	<NA>
## PF-543	<NA>
## PF-3758309	Cytoskeletal Signaling
## AZD5438	Cell Cycle
## R547	Cell Cycle
## A-674563	<NA>
## Dinaciclib (SCH727965)	Cell Cycle
## P276-00	Cell Cycle
## Flavopiridol HCl	Cell Cycle
## PHA-767491	Cell Cycle
## PF-00562271	<NA>
## PF-562271	<NA>
## IKK-16 (IKK Inhibitor VII)	<NA>
## LDK378	<NA>
## Vacquinol-1	<NA>
## GSK2126458 (GSK458)	<NA>
## Torin 2	<NA>
## AZD3463	<NA>
## AP26113	<NA>
## SC1	<NA>
## SGI-7079	<NA>
## PF-431396	<NA>
## TAE226 (NVP-TAE226)	<NA>
## Milciclib (PHA-848125)	<NA>
## TG101209	<NA>
## TG101348 (SAR302503)	<NA>
## GSK690693	<NA>
## GDC-0068	<NA>
## MK-2206 2HCl	<NA>
## AZD5363	<NA>

Drug-Induced Phenotypes

```

## Uprosertib (GSK2141795) <NA>
## BMS-536924 <NA>
## OSI-906 (Linsitinib) <NA>
## ZSTK474 <NA>
## KU-0063794 <NA>
## WYE-354 <NA>
## Temsirolimus (CCI-779, NSC 683864) <NA>
## Everolimus (RAD001) <NA>
## Ridaforolimus (Deforolimus, MK-8669) <NA>
## GDC-0980 (RG7422) PI3K/Akt/mTOR
## INK 128 (MLN0128) PI3K/Akt/mTOR
## VS-5584 (SB2343) PI3K/Akt/mTOR
## PF-04691502 PI3K/Akt/mTOR
## PP121 <NA>
## CC-223 PI3K/Akt/mTOR
## PP242 PI3K/Akt/mTOR
## AZD2014 PI3K/Akt/mTOR
## GDC-0349 PI3K/Akt/mTOR
## OSI-027 PI3K/Akt/mTOR
## AZD8055 PI3K/Akt/mTOR
## CH5132799 PI3K/Akt/mTOR
## WYE-125132 (WYE-132) PI3K/Akt/mTOR
## CO-1686 (AVL-301) <NA>
## A66 PI3K/Akt/mTOR
## BYL719 PI3K/Akt/mTOR
## Akti-1/2 PI3K/Akt/mTOR
## WAY-600 PI3K/Akt/mTOR
## ETP-46464 PI3K/Akt/mTOR
## CCT128930 PI3K/Akt/mTOR
## GSK2334470 PI3K/Akt/mTOR
## Nintedanib (BIBF 1120)_uncertain <NA>
## WIKI4 Stem Cells & Wnt
## Axitinib <NA>
## PHA-665752 <NA>
## PD173074 <NA>
## AT13148 <NA>
## AT7867 <NA>
## Linifanib (ABT-869) <NA>
## PD184352 (CI-1040) <NA>
## U0126-EtOH <NA>
## Dovitinib (TKI-258, CHIR-258) <NA>
## Sunitinib Malate <NA>
## Cediranib (AZD2171) <NA>
## SGI-1776 free base <NA>
## GSK2636771 <NA>
## RepSox <NA>
## K02288 <NA>
## TWS119 <NA>
## ICG-001 Stem Cells & Wnt
## PRI-724 Stem Cells & Wnt
## AZD4547 <NA>

```

Drug-Induced Phenotypes

## CEP-33779	<NA>
## Y0-01027	<NA>
## WZ4003	<NA>
## AVL-292	<NA>
## ASP3026	<NA>
## Entrectinib (RXDX-101)	<NA>
## BI0	<NA>
## Semaxanib (SU5416)	<NA>
## SU6656	Angiogenesis
## DCC-2036 (Rebastinib)	Angiogenesis
## Purvalanol A	<NA>
## Dabrafenib (GSK2118436)	<NA>
## PP1	Angiogenesis
## PP2	Angiogenesis
## WH-4-023	Angiogenesis
## Dasatinib	Angiogenesis
## PD173955	Angiogenesis
## Ponatinib (AP24534)	Angiogenesis
## PRT062607 (P505-15, BIIB057) HCl	Angiogenesis
## LDN-214117	<NA>
## LGK-974	Stem Cells & Wnt
## AZ 960	<NA>
## DASA-58	<NA>
## FH535	Stem Cells & Wnt
## Mubritinib (TAK 165)	<NA>
## VPS34-IN1	<NA>
## IMD 0354	<NA>
## Tyrphostin 9	<NA>
## Pazopanib	<NA>
## Sorafenib	<NA>
## SU11274	<NA>
## AG-1024	<NA>
## Amuvatinib (MP-470)	<NA>
## Sorafenib Tosylate	<NA>
## Foretinib (GSK1363089)	<NA>
## SNS-314 Mesylate	<NA>
## CYC116	<NA>
## VX-680 (Tozastertib, MK-0457)	<NA>
## BX-912	<NA>
## AZD1480	<NA>
## Danusertib (PHA-739358)	<NA>
## Quizartinib (AC220)	<NA>
## AEE788 (NVP-AEE788)	<NA>
## WZ3146	<NA>
## ZM 447439	<NA>
## Barasertib (AZD1152-HQPA)	<NA>
## PHA-680632	<NA>
## Aurora A Inhibitor I	<NA>
## OSU-03012 (AR-12)	<NA>
## MK-5108 (VX-689)	<NA>
## HTH-01-015	<NA>

Drug-Induced Phenotypes

## MK-8745	<NA>
## TPCA-1	<NA>
## BMS-345541	<NA>
## Dovitinib (TKI-258) Dilactic Acid	<NA>
## GSK1838705A	<NA>
## NVP-BSK805 2HCl	<NA>
## LY2784544	<NA>
## SU9516	<NA>
## CNX-2006	Protein Tyrosine Kinase
## Dacomitinib (PF299804, PF299)	Protein Tyrosine Kinase
## JNK-IN-8	<NA>
## AZD9291	Protein Tyrosine Kinase
## GZD824	<NA>
## ERK5-IN-1	<NA>
## XMD8-92	<NA>
## LY2835219	Cell Cycle
## CP-673451	<NA>
## Crenolanib (CP-868596)	<NA>
## PHA-793887	Cell Cycle
## Ro 31-8220 Mesylate	<NA>
## Ro3280	Cell Cycle
## LY2603618	Cell Cycle
## PF-477736	Cell Cycle
## TAK-632	<NA>
## LDC000067	Cell Cycle
## Fostamatinib (R788)	<NA>
## TAK-901	Cell Cycle
## Bosutinib (SKI-606)	<NA>
## AST-1306	Protein Tyrosine Kinase
## Imatinib Mesylate (STI571)	Protein Tyrosine Kinase
## AZD8931 (Sapitinib)	Protein Tyrosine Kinase
## Gefitinib (ZD1839)	Protein Tyrosine Kinase
## Saracatinib (AZD0530)	<NA>
## OSI-420	Protein Tyrosine Kinase
## Pelitinib (EKB-569)	Protein Tyrosine Kinase
## AZD3759	Protein Tyrosine Kinase
## PD168393	Protein Tyrosine Kinase
## AG-1478 (Tyrphostin AG-1478)	Protein Tyrosine Kinase
## ZM 306416	Protein Tyrosine Kinase
## AC480 (BMS-599626)	<NA>
## Ibrutinib (PCI-32765)	<NA>
## ONO-4059	<NA>
## LY2090314	PI3K/Akt/mTOR
## AZD2858	PI3K/Akt/mTOR
## CHIR-98014	PI3K/Akt/mTOR
## AZD1080	PI3K/Akt/mTOR
## CHIR-99021 (CT99021) HCl	PI3K/Akt/mTOR
## Enzastaurin (LY317615)	<NA>
## 1-Azakenpaullone	PI3K/Akt/mTOR
## CHIR-99021 (CT99021)	PI3K/Akt/mTOR
## Sotrastaurin	TGF-beta/Smad

Drug-Induced Phenotypes

	TGF-beta/Smad	
## GF109203X	TGF-beta/Smad	
## Go 6983	TGF-beta/Smad	
## AZ 628	<NA>	
## MLN2480	<NA>	
## CX-6258 HCl	<NA>	
## Piceatannol	<NA>	
## PQ 401	<NA>	
## Tyrophostin AG 879	<NA>	
##	EnrichedTarget	Lethality
## Ulixertinib (BVD-523, VRT752271)	<NA>	0
## TAK-733	mek	6
## Trametinib (GSK1120212)	mek	5
## Cobimetinib (GDC-0973, RG7420)	mek	4
## MEK162 (ARRY-162, ARRY-438162)	mek	2
## PD0325901	mek	4
## Refametinib (RDEA119, Bay 86-9766)	mek	4
## AZD8330	mek	4
## Pimasertib (AS-703026)	mek	3
## RAF265 (CHIR-265)	<NA>	6
## Selumetinib (AZD6244)	mek	2
## PD318088	mek	2
## VX-11e	<NA>	1
## NVP-BHG712	<NA>	0
## YM201636	<NA>	0
## Lapatinib (GW-572016) Ditosylate	<NA>	0
## PF-573228	<NA>	0
## BMS-265246	<NA>	0
## KU-60019	<NA>	0
## Triciribine	<NA>	0
## CUDC-101	<NA>	5
## GSK461364	plk	0
## AZD7762	<NA>	7
## Rigosertib (ON-01910)	plk	5
## BI 2536	plk	3
## JNJ-7706621	<NA>	6
## PIK-93	<NA>	0
## PF-4708671	<NA>	0
## Flavopiridol (Alvocidib)	cdk	10
## AT7519	cdk	8
## SNS-032 (BMS-387032)	cdk	9
## Staurosporine_500nM	<NA>	4
## BMS-754807	<NA>	3
## LY3023414	<NA>	2
## Hesperadin	aurora kinase	3
## KW-2449	aurora kinase	2
## Crizotinib (PF-02341066)	<NA>	2
## ENMD-2076	aurora kinase	1
## HMN-214	<NA>	1
## BKM120 (NVP-BKM120, Buparlisib)	<NA>	2
## NVP-AEW541	<NA>	4
## Alisertib (MLN8237)	aurora kinase	0

Drug-Induced Phenotypes

## GSK2292767	<NA>	0
## BGT226 (NVP-BGT226)	<NA>	12
## Bortezomib	<NA>	14
## CUDC-907	<NA>	12
## CHIR-124	<NA>	9
## Pacritinib (SB1518)	jak	14
## AT9283	jak	10
## Irinotecan / SN-38	<NA>	14
## JNK Inhibitor IX	<NA>	0
## Tivantinib (ARQ 197)	c-met	1
## HS-173	<NA>	0
## Wnt agonist 1	<NA>	0
## FRAX597	<NA>	0
## Tepotinib (EMD 1214063)	c-met	2
## Golvatinib (E7050)	c-met	3
## KX2-391	<NA>	1
## VE-822	atm/atr	2
## AZ20	atm/atr	1
## AZD6738	atm/atr	2
## AMG-900	<NA>	0
## PF-543	<NA>	0
## PF-3758309	<NA>	5
## AZD5438	cdk	9
## R547	cdk	10
## A-674563	cdk	11
## Dinaciclib (SCH727965)	cdk	9
## P276-00	cdk	10
## Flavopiridol HCl	cdk	8
## PHA-767491	cdk	5
## PF-00562271	fak	5
## PF-562271	fak	4
## IKK-16 (IKK Inhibitor VII)	<NA>	4
## LDK378	alk	4
## Vacquinol-1	<NA>	2
## GSK2126458 (GSK458)	<NA>	4
## Torin 2	<NA>	6
## AZD3463	alk	6
## AP26113	alk	8
## SC1	<NA>	5
## SGI-7079	<NA>	3
## PF-431396	fak	1
## TAE226 (NVP-TAE226)	fak	2
## Milciclib (PHA-848125)	<NA>	6
## TG101209	jak	3
## TG101348 (SAR302503)	jak	7
## GSK690693	akt	0
## GDC-0068	akt	0
## MK-2206 2HCl	akt	0
## AZD5363	akt	0
## Urosertib (GSK2141795)	akt	0
## BMS-536924	<NA>	0

Drug-Induced Phenotypes

## OSI-906 (Linsitinib)	<NA>	0
## ZSTK474	<NA>	0
## KU-0063794	mtor	0
## WYE-354	mtor	0
## Temsirolimus (CCI-779, NSC 683864)	mtor	0
## Everolimus (RAD001)	mtor	0
## Ridaforolimus (Deforolimus, MK-8669)	mtor	0
## GDC-0980 (RG7422)	mtor	0
## INK 128 (MLN0128)	mtor	0
## VS-5584 (SB2343)	pi3k	0
## PF-04691502	mtor	0
## PP121	mtor	7
## CC-223	mtor	0
## PP242	mtor	0
## AZD2014	mtor	0
## GDC-0349	mtor	0
## OSI-027	mtor	0
## AZD8055	mtor	0
## CH5132799	mtor	0
## WYE-125132 (WYE-132)	mtor	0
## CO-1686 (AVL-301)	<NA>	0
## A66	pi3k	0
## BYL719	pi3k	0
## Akti-1/2	<NA>	0
## WAY-600	mtor	0
## ETP-46464	mtor	0
## CCT128930	<NA>	0
## GSK2334470	<NA>	0
## Nintedanib (BIBF 1120)_uncertain	vegfr	0
## WIKI4	wnt/beta-catenin	0
## Axitinib	vegfr	0
## PHA-665752	<NA>	0
## PD173074	vegfr	0
## AT13148	<NA>	0
## AT7867	<NA>	0
## Linifanib (ABT-869)	vegfr	0
## PD184352 (CI-1040)	<NA>	0
## U0126-EtOH	<NA>	0
## Dovitinib (TKI-258, CHIR-258)	vegfr	0
## Sunitinib Malate	vegfr	0
## Cediranib (AZD2171)	vegfr	0
## SGI-1776 free base	<NA>	2
## GSK2636771	<NA>	0
## RepSox	tgf-beta/smad	0
## K02288	tgf-beta/smad	0
## TWS119	<NA>	0
## ICG-001	wnt/beta-catenin	0
## PRI-724	wnt/beta-catenin	0
## AZD4547	fgfr	0
## CEP-33779	<NA>	0
## Y0-01027	<NA>	0

Drug-Induced Phenotypes

## WZ4003	<NA>	0
## AVL-292	<NA>	0
## ASP3026	<NA>	0
## Entrectinib (RXDX-101)	<NA>	0
## BIO	<NA>	0
## Semaxanib (SU5416)	<NA>	0
## SU6656	src	0
## DCC-2036 (Rebastinib)	<NA>	0
## Purvalanol A	<NA>	0
## Dabrafenib (GSK2118436)	<NA>	0
## PP1	src	0
## PP2	src	0
## WH-4-023	src	0
## Dasatinib	src	0
## PD173955	bcr-abl	0
## Ponatinib (AP24534)	bcr-abl	1
## PRT062607 (P505-15, BIIB057) HCl	<NA>	0
## LDN-214117	tgf-beta/smad	0
## LGK-974	wnt/beta-catenin	0
## AZ 960	<NA>	1
## DASA-58	<NA>	0
## FH535	wnt/beta-catenin	0
## Mubritinib (TAK 165)	<NA>	0
## VPS34-IN1	<NA>	0
## IMD 0354	<NA>	3
## Tyrphostin 9	<NA>	0
## Pazopanib	<NA>	0
## Sorafenib	<NA>	0
## SU11274	<NA>	0
## AG-1024	<NA>	0
## Amuvatinib (MP-470)	<NA>	0
## Sorafenib Tosylate	<NA>	0
## Foretinib (GSK1363089)	<NA>	0
## SNS-314 Mesylate	aurora kinase	0
## CYC116	aurora kinase	0
## VX-680 (Tozasertib, MK-0457)	aurora kinase	0
## BX-912	<NA>	0
## AZD1480	<NA>	1
## Danusertib (PHA-739358)	aurora kinase	0
## Quizartinib (AC220)	<NA>	0
## AEE788 (NVP-AEE788)	<NA>	1
## WZ3146	<NA>	0
## ZM 447439	aurora kinase	0
## Barasertib (AZD1152-HQPA)	aurora kinase	0
## PHA-680632	aurora kinase	0
## Aurora A Inhibitor I	aurora kinase	0
## OSU-03012 (AR-12)	<NA>	0
## MK-5108 (VX-689)	<NA>	0
## HTH-01-015	<NA>	0
## MK-8745	<NA>	0
## TPCA-1	<NA>	0

Drug-Induced Phenotypes

## BMS-345541	<NA>	0
## Dovitinib (TKI-258) Dilactic Acid	<NA>	0
## GSK1838705A	<NA>	0
## NVP-BSK805 HCl	<NA>	0
## LY2784544	<NA>	0
## SU9516	<NA>	0
## CNX-2006	egfr	1
## Dacomitinib (PF299804, PF299)	egfr	1
## JNK-IN-8	<NA>	1
## AZD9291	egfr	1
## GZD824	<NA>	1
## ERK5-IN-1	<NA>	0
## XMD8-92	<NA>	0
## LY2835219	cdk	0
## CP-673451	<NA>	0
## Crenolanib (CP-868596)	<NA>	0
## PHA-793887	cdk	0
## Ro 31-8220 Mesylate	<NA>	2
## Ro3280	<NA>	0
## LY2603618	<NA>	0
## PF-477736	<NA>	1
## TAK-632	<NA>	0
## LDC000067	cdk	0
## Fostamatinib (R788)	<NA>	0
## TAK-901	<NA>	1
## Bosutinib (SKI-606)	src	1
## AST-1306	egfr	0
## Imatinib Mesylate (STI571)	<NA>	0
## AZD8931 (Sapitinib)	egfr	0
## Gefitinib (ZD1839)	egfr	0
## Saracatinib (AZD0530)	src	1
## OSI-420	egfr	1
## Pelitinib (EKB-569)	egfr	2
## AZD3759	egfr	0
## PD168393	egfr	0
## AG-1478 (Tyrphostin AG-1478)	egfr	0
## ZM 306416	<NA>	0
## AC480 (BMS-599626)	her2	0
## Ibrutinib (PCI-32765)	src	0
## ONO-4059	<NA>	0
## LY2090314	gsk-3	0
## AZD2858	gsk-3	0
## CHIR-98014	gsk-3	0
## AZD1080	gsk-3	0
## CHIR-99021 (CT99021) HCl	<NA>	0
## Enzastaurin (LY317615)	<NA>	0
## 1-Azakenpaullone	gsk-3	0
## CHIR-99021 (CT99021)	gsk-3	0
## Sotрастaurин	pkc	0
## GF109203X	pkc	0
## Go 6983	pkc	0

Drug-Induced Phenotypes

## AZ 628	<NA>	0
## MLN2480	<NA>	0
## CX-6258 HCl	<NA>	0
## Piceatannol	<NA>	0
## PQ 401	<NA>	0
## Tyrphostin AG 879	her2	0

Generate a force-directed graph from these similarities

```
# Set up adjacency matrix
adjmatrix = d
adjmatrix[adjmatrix > 45] = 0

# Resort matrix to make sure annotated nodes are above unannotated ones
sort_order = order(annotation$Target, na.last = FALSE)
adjmatrix = adjmatrix[sort_order, sort_order]

# Remove drugs without any connections
sel_indices = colSums(adjmatrix != 0) != 0
adjmatrix = adjmatrix[sel_indices, sel_indices]

# Convert to cosine distance: D = 1 - cos(angle)
# adjmatrix = 1 - cos(adjmatrix * pi / 180)

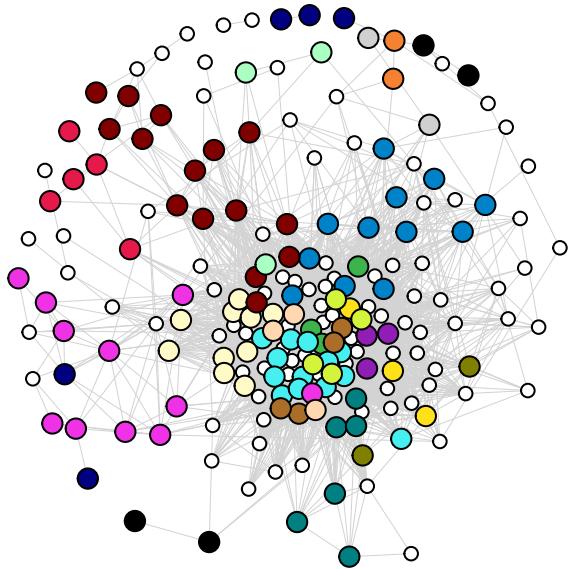
# Generate network
adjmatrix = Matrix::Matrix(data = adjmatrix, sparse = TRUE)
net = graph_from_adjacency_matrix(
  adjmatrix = adjmatrix, mode = "undirected", diag = FALSE, weighted = TRUE)

# Assign targets and colors
V(net)$target = annotation[match(V(net)$name, rownames(annotation)), "Target"]
V(net)$color = V(net)$target
for(target in na.omit(unique(V(net)$target))) {
  regex = paste0("\\b", target, "\\b")
  V(net)$color = gsub(regex, anno_colorScale$Target[target], V(net)$color)}
V(net)$color[is.na(V(net)$color)] = "white"

# Assign size
V(net)$size = 5
V(net)[!is.na(V(net)$target)]$size = 7.5

# Create plot
e = get.edgelist(net, names = FALSE)
l = qgraph.layout.fruchtermanreingold(
  edgelist = e, vcount = vcount(net),
  area=20*(vcount(net)^2), repulse.rad=(vcount(net)^3.1))
# l = layout_with_graphopt(graph = net, niter = 1000, mass = 100)
plot.igraph(net, vertex.label = NA, layout=l,
            edge.width = 0.5, edge.color = "lightgray")
```

Drug-Induced Phenotypes



```
if(save_images) {  
  pdf(file = file.path(img_out_dir, "EffectVectorNetwork_allLines.pdf"),  
       width = 7, height = 7, useDingbats = FALSE)  
  plot.igraph(net, vertex.label = NA, layout=l,  
             edge.width = 0.5, edge.color = "lightgray")  
  dev.off()}
```

4 Clinical Cancer Panel

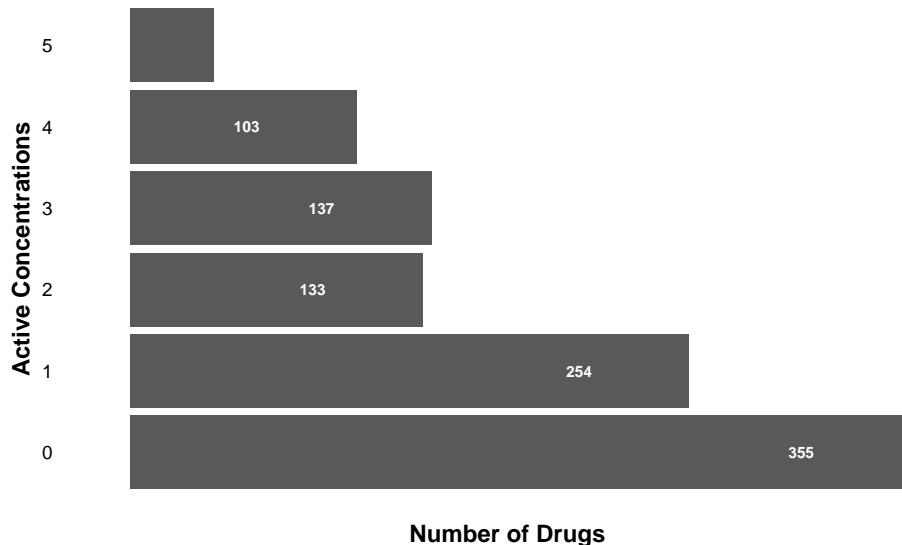
```
sel_indices = !is.na(drug_effect_metadata$Concentration)  
profiles = drug_effect_profiles[sel_indices, ]  
metadata = drug_effect_metadata[sel_indices, ]  
# Sort for convenience  
order_indices = order(rownames(profiles))  
profiles = profiles[order_indices, ]  
metadata = metadata[order_indices, ]
```

4.1 Drugs Active at All Concentrations

```
sig_concentrations = aggregate(  
  x = metadata$AUC_Mean >= auc_thresh,  
  by = list("Drug" = metadata$Drug,  
           "Line" = metadata$Line),  
  FUN = sum)  
  
sig_conc_table = as.data.frame(table(sig_concentrations$x))  
sig_conc_table$TextLoc = sig_conc_table$Freq - 50  
ggplot(data = sig_conc_table) +
```

Drug-Induced Phenotypes

```
geom_col(mapping = aes(x = Var1, y = Freq)) + theme_vignette() +
  geom_text(mapping = aes(x = Var1, y = TextLoc, label = Freq),
            color = "white", fontface = "bold", size = 3) +
  xlab("Active Concentrations") + ylab("Number of Drugs") + coord_flip() +
  theme(axis.line = element_blank(), axis.text.x = element_blank(),
        axis.ticks = element_blank())
```



```
if(save_images) ggsave(
  filename = file.path(img_out_dir, "Barplot_NumActiveConcentrations.pdf"),
  width = 2.5, height = 3)

# The number of concentrations at which drugs should be active
conc_thresh = 5
sig_concentrations = sig_concentrations[
  sig_concentrations$x >= conc_thresh, ]

# Create heatmap for each entry
for(ii in seq_len(nrow(sig_concentrations))) {
  drug = sig_concentrations[ii, "Drug"]
  line = sig_concentrations[ii, "Line"]
  sel_indices = metadata$Line == line & metadata$Drug == drug
  entry = profiles[sel_indices, ]
  entry_md = metadata[sel_indices, ]

  angles = get_angles(entry)
  d = acos(angles)*180/pi
  hc = hclust(as.dist(d), method = "ward.D2")

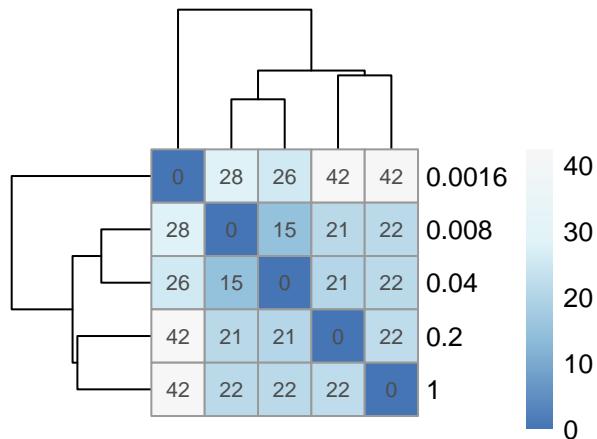
  hm_colorscale = colorRampPalette(
    rev(c("#f7f7f7", "#E0F3F8", "#91BFDB", "#4575B4")))(150)

  pheatmap(
```

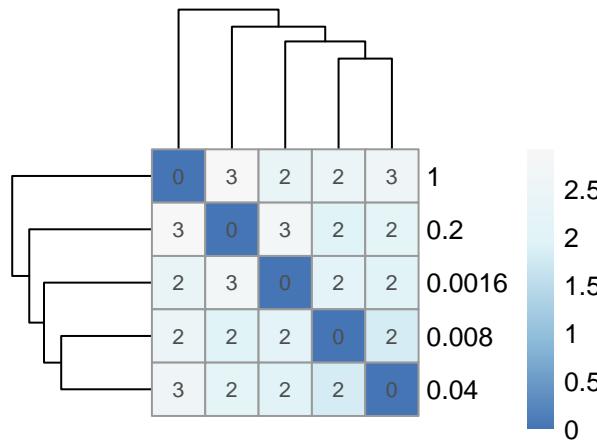
Drug-Induced Phenotypes

```
d, cluster_rows = hc, cluster_cols = hc, color = hm_colorscale,  
display_numbers = round(d), cellwidth = 20, cellheight = 20,  
labels_row = entry_md$Concentration, show_colnames = FALSE,  
main = paste0(line, " - ", drug))  
}
```

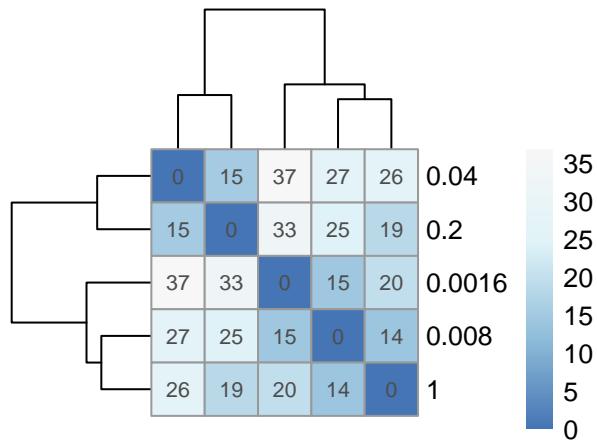
D004T01 – Irinotecan / SN-38



D007T01 – Doxorubicin

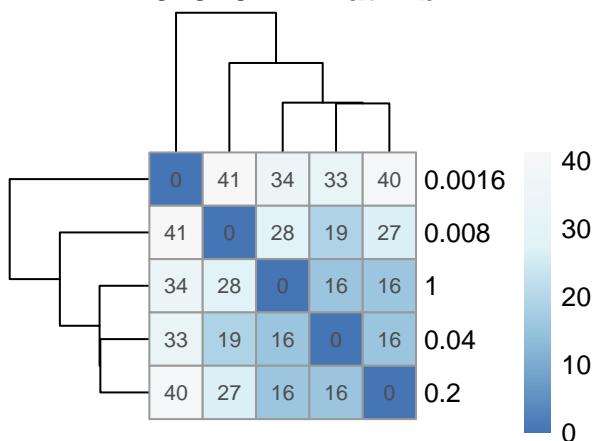


D007T01 – Volasertib

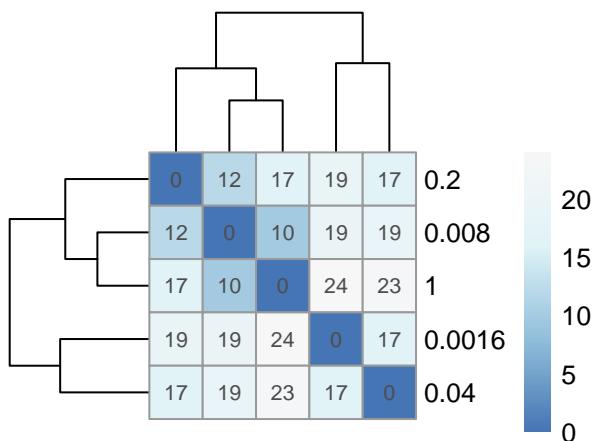


Drug-Induced Phenotypes

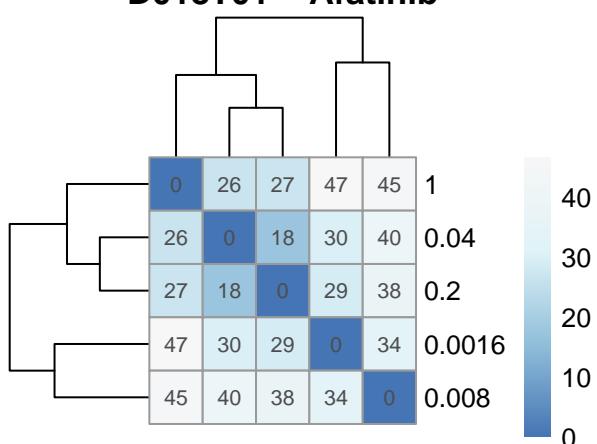
D013T01 – Afatinib



D013T01 – Irinotecan / SN-38

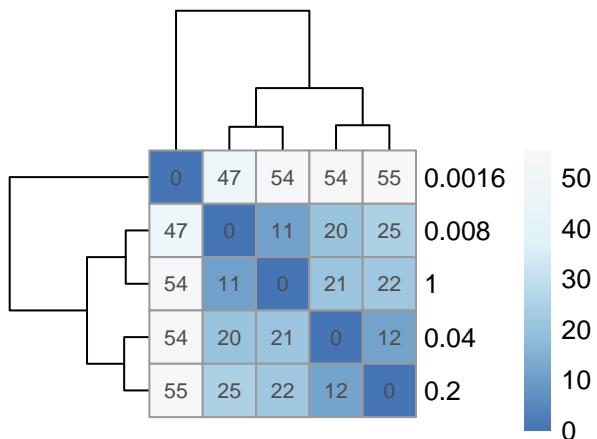


D018T01 – Afatinib

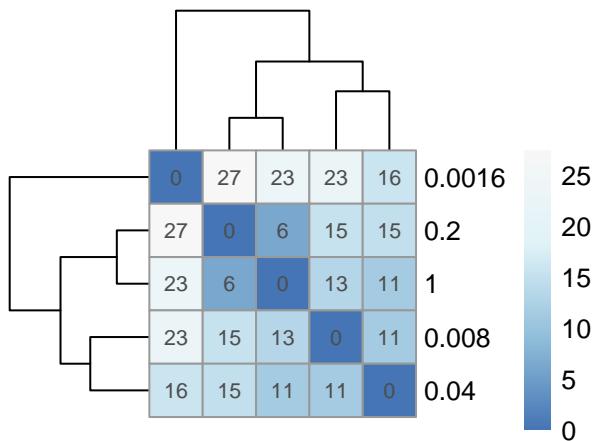


Drug-Induced Phenotypes

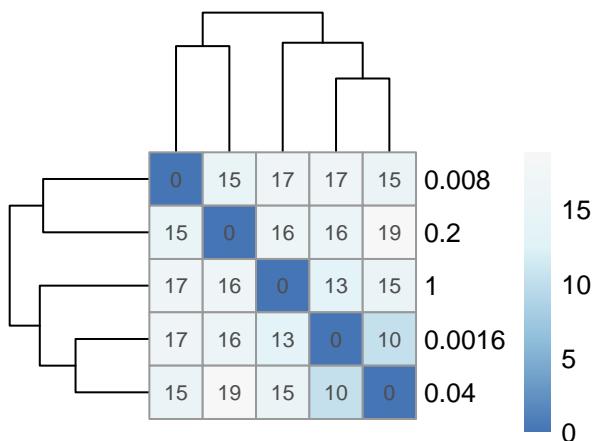
D018T01 – Irinotecan / SN-38



D019T01 – Irinotecan / SN-38

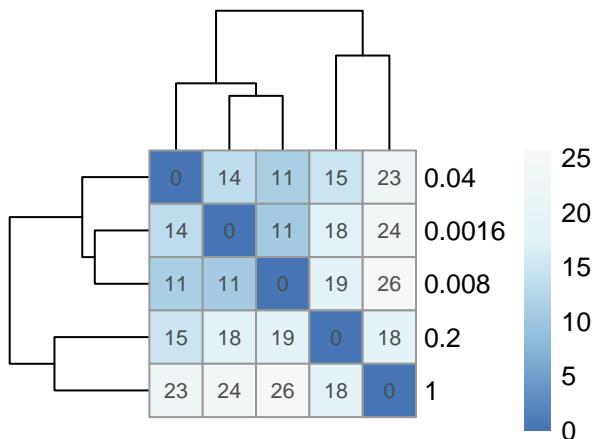


D020T01 – Docetaxel

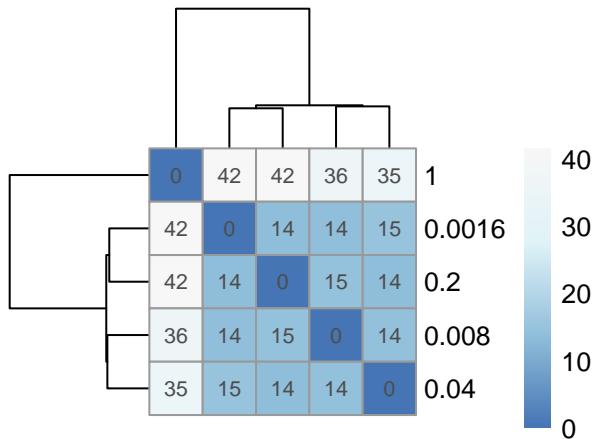


Drug-Induced Phenotypes

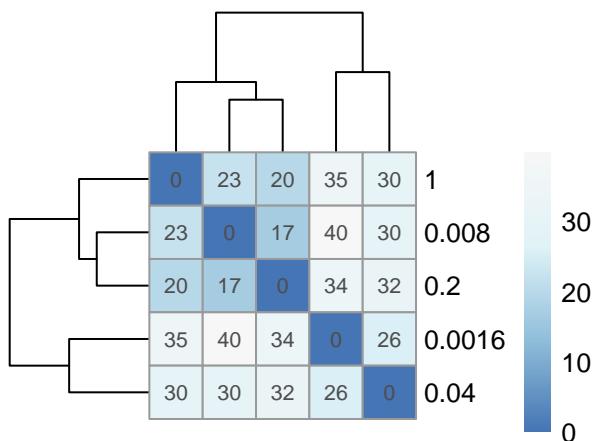
D020T01 – Irinotecan / SN-38



D020T01 – Volasertib

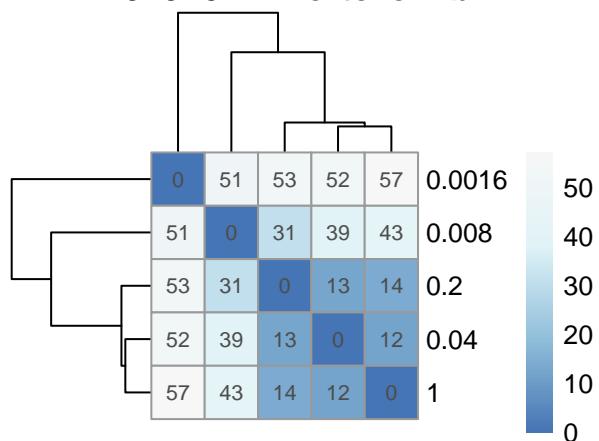


D020T02 – Afatinib

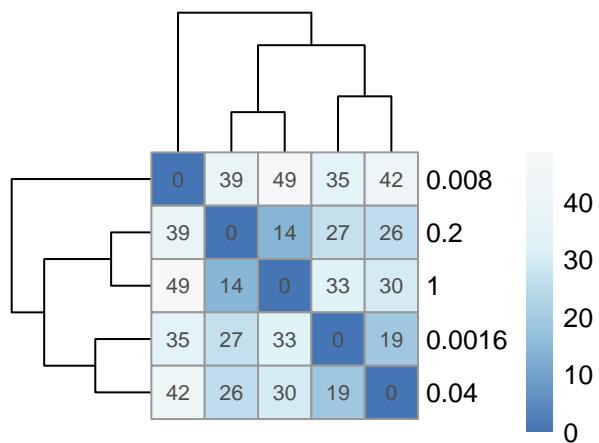


Drug-Induced Phenotypes

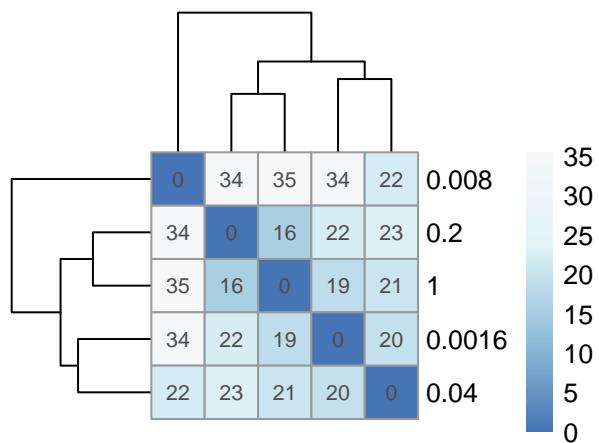
D020T02 – Bortezomib



D020T02 – Irinotecan / SN-38

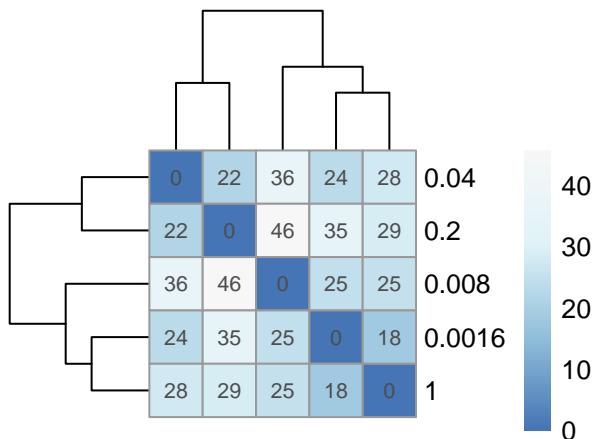


D020T02 – Irinotecan mit 5-FU

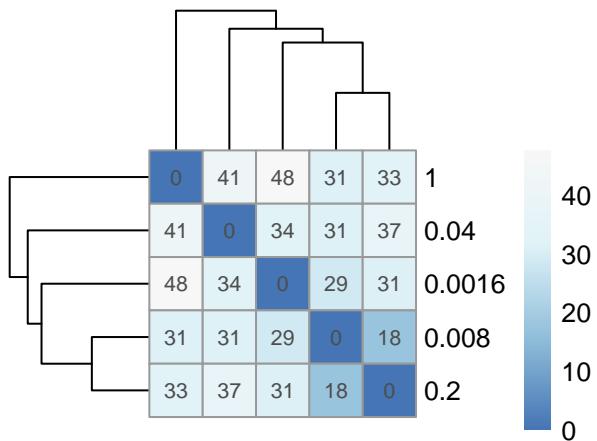


Drug-Induced Phenotypes

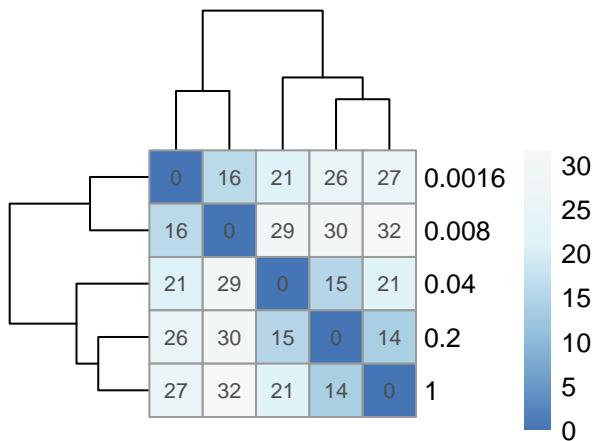
D020T02 – Trametinib



D020T02 – Volasertib

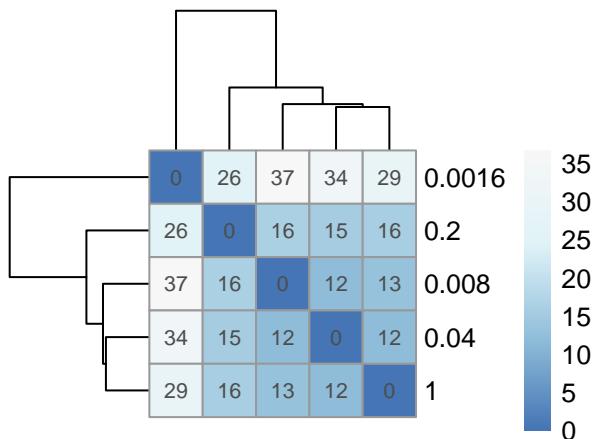


D021T01 – Dasatinib

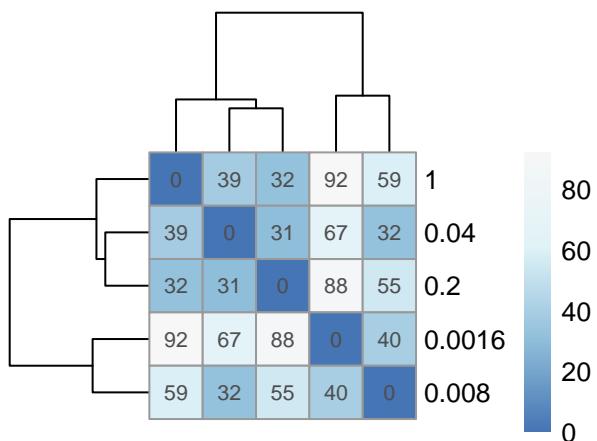


Drug-Induced Phenotypes

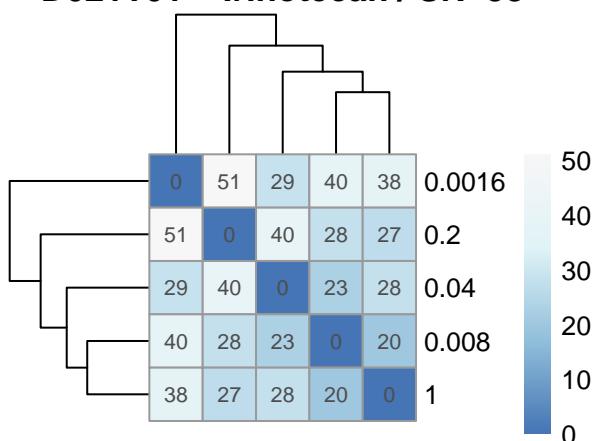
D021T01 – Docetaxel



D021T01 – Doxorubicin

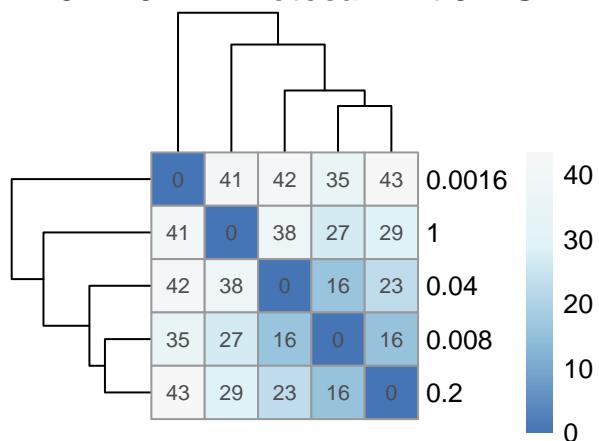


D021T01 – Irinotecan / SN-38

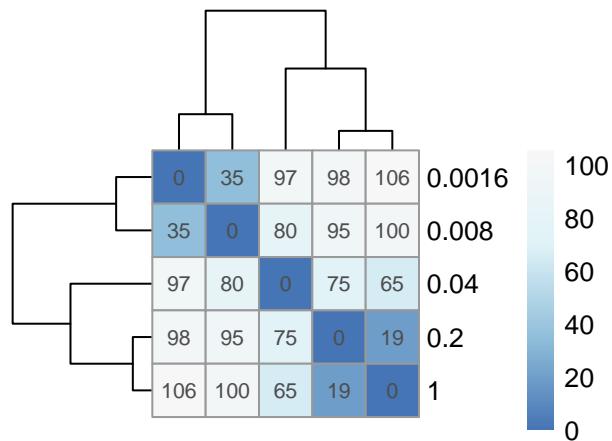


Drug-Induced Phenotypes

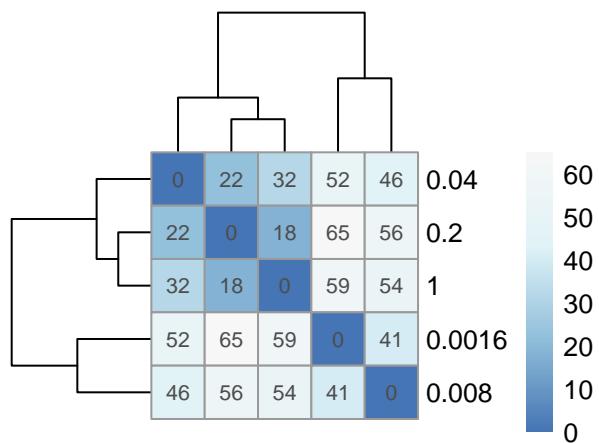
D021T01 – Irinotecan mit 5-FU

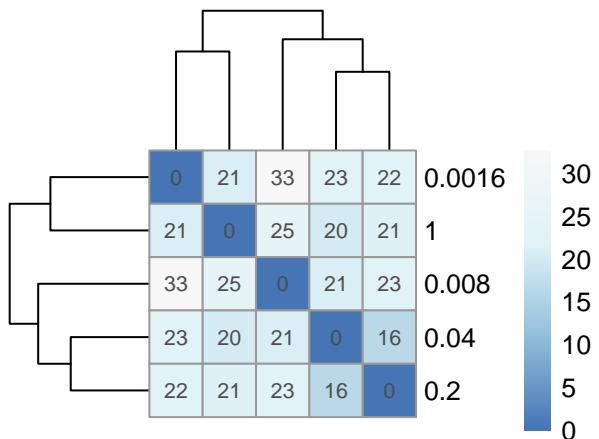
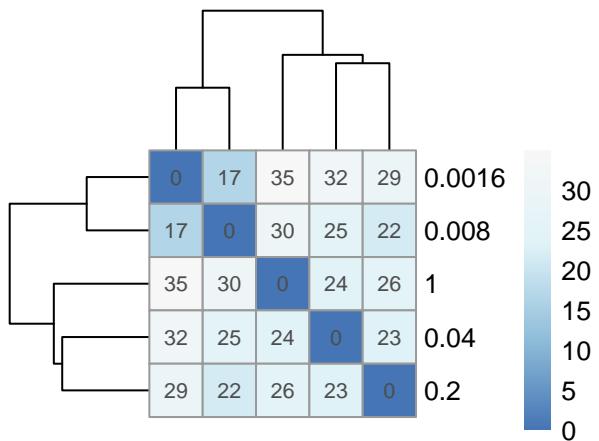
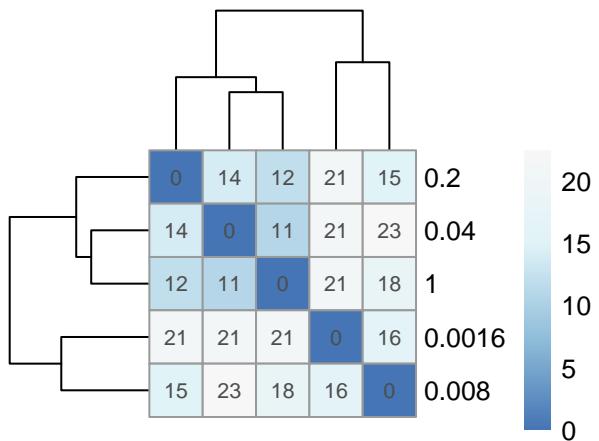


D021T01 – Panobinostat



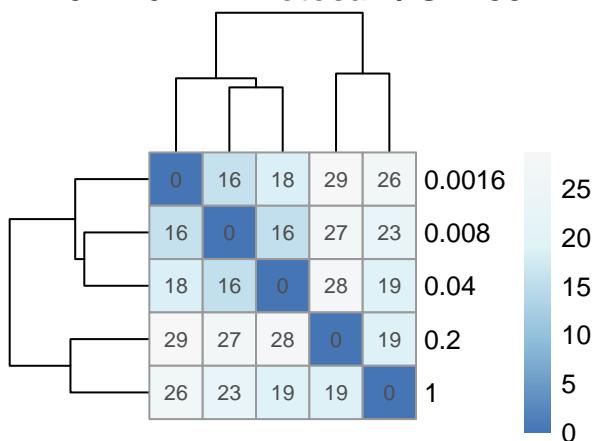
D021T01 – Trametinib



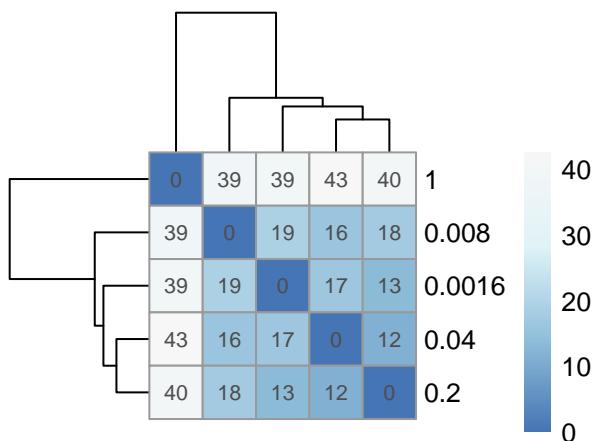
D021T01 – Vinblastin**D021T01 – Volasertib****D022T01 – Docetaxel**

Drug-Induced Phenotypes

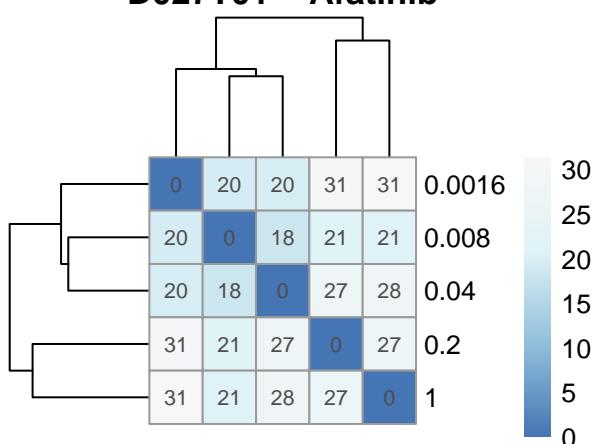
D022T01 – Irinotecan / SN-38



D022T01 – Volasertib

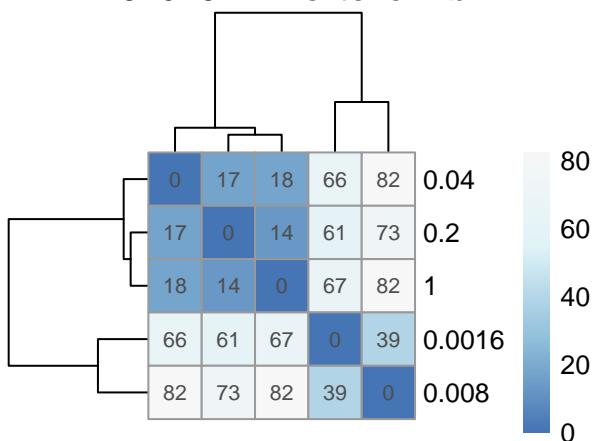


D027T01 – Afatinib

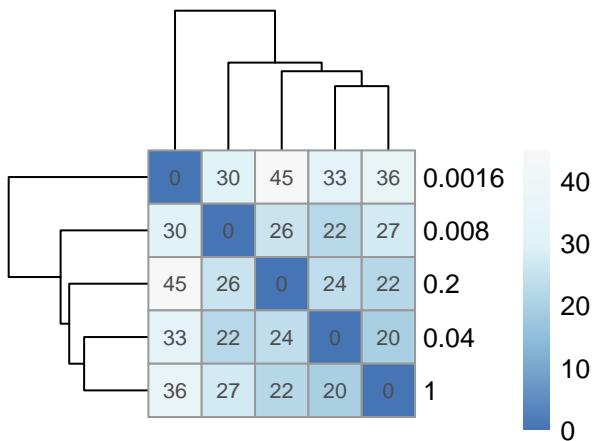


Drug-Induced Phenotypes

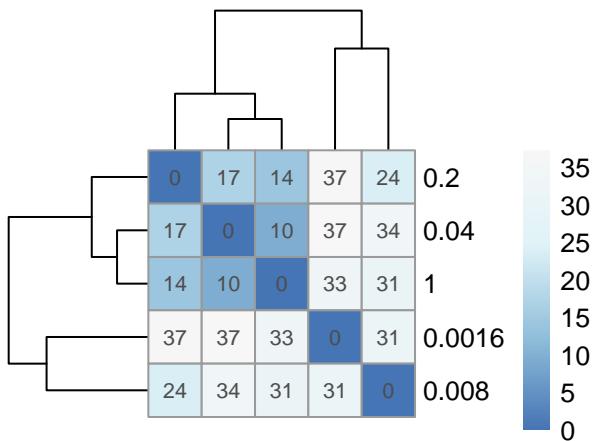
D046T01 – Bortezomib



D046T01 – Irinotecan / SN-38

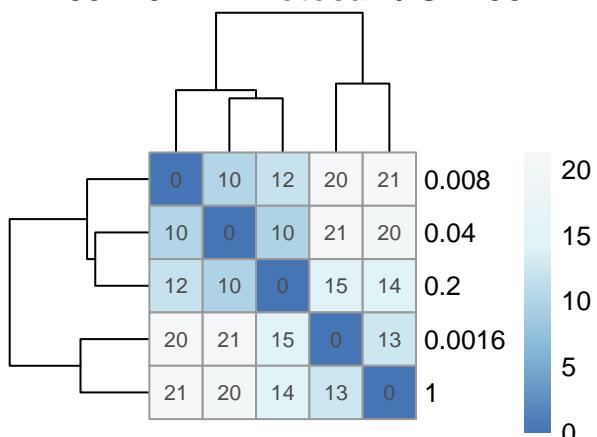


D054T01 – Bortezomib

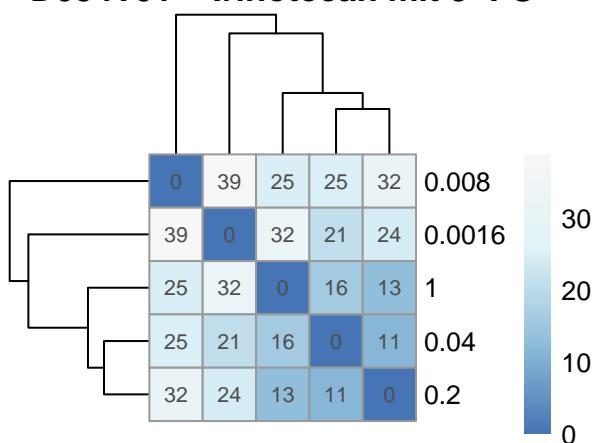


Drug-Induced Phenotypes

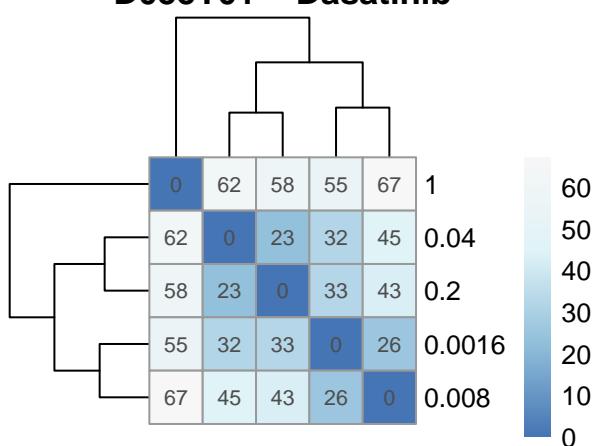
D054T01 – Irinotecan / SN-38



D054T01 – Irinotecan mit 5-FU

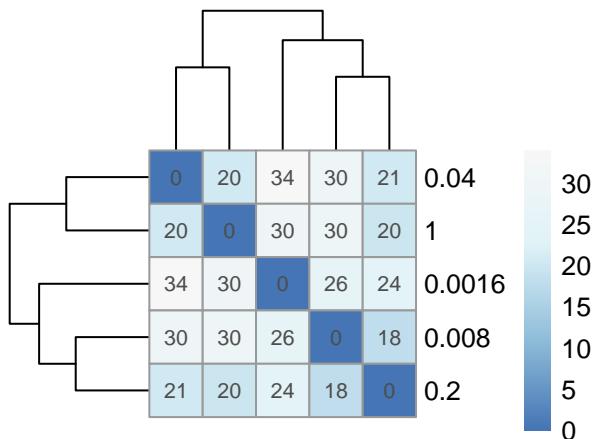


D055T01 – Dasatinib



Drug-Induced Phenotypes

D055T01 – Docetaxel



D055T01 – Irinotecan / SN-38

