Agentic AI for Adaptive Pharmacogenomic Biomarker Discovery in Cancer

Applications in Preclinical Biomarker Discovery

Pharmacogenomic datasets provide a wealth of biological information across various experiments and sequencing technologies for a large collection of preclinical samples. While this makes for an ideal data source for preliminary biomarker discovery, the data container structure can be difficult to navigate. An agentic framework, provided with tools from the PharmacoGx package, can expedite bioinformatic analysis.

Case scenario 1:

Gene expression-based subtypes of breast cancer are widely studied to be associated with response to treatment.

The following dataset was downloaded from orcestra.ca. The dataset is stored as a `PharmacoSet` object from the PharmacoGx package. The dataset contains 74 breast cancer samples with 9756 drug response experiments.

```
gray <- readRDS(here("./data/rawdata/PSet_GRAY2017.rds")) |> updateObject()
gray
```

```
## <PharmacoSet>
## Name: GRAY
## Date Created: Thu Apr  6 16:50:15 2023
## Number of samples:  74
## Molecular profiles:
## Kallisto_0.46.1.rnaseq :
##   Dim: 60662, 64
## Kallisto_0.46.1.rnaseq.counts :
##   Dim: 60662, 64
## Kallisto_0.46.1.isoforms :
##   Dim: 227912, 64
## Kallisto_0.46.1.isoforms.counts :
##   Dim: 227912, 64
## Treatment response: Drug pertubation:
##    Please look at pertNumber(cSet) to determine number of experiments  for each drug-sample combinati
## Drug sensitivity:
##    Number of Experiments:  9756
##    Please look at sensNumber(cSet) to determine number of  experiments for each drug-sample combinati
```

Task: compute pam50 subtypes

```
# load in genefu PAM50 subtyping model
data(pam50.robust)
```

```
# get gene expression
gray_rna <- summarizeMolecularProfiles(
  gray,
  mDataType = "Kallisto_0.46.1.rnaseq"
) |> suppressMessages()
```

```
##   |                                                                      |
```

```
rna <- t(assay(gray_rna)) |> as.data.frame()

# create metadata file
meta <- gray_rna@elementMetadata[,c("gene_id", "gene_name")] |> as.data.frame()
colnames(meta) <- c("Ensembl", "Gene.Symbol")
colnames(rna) <- meta$Gene.Symbol[match(colnames(rna), meta$Ensembl)]

# get subtype predictions for PAM50
pam50 <- molecular.subtyping(
  sbt.model = "pam50",
  data = rna,
  annot = meta,
  do.mapping = FALSE
)
gray_pam50 <- cbind(
  Subtype = as.character(pam50$subtype),
  as.data.frame(pam50$subtype.proba)
)
head(gray_pam50)
```

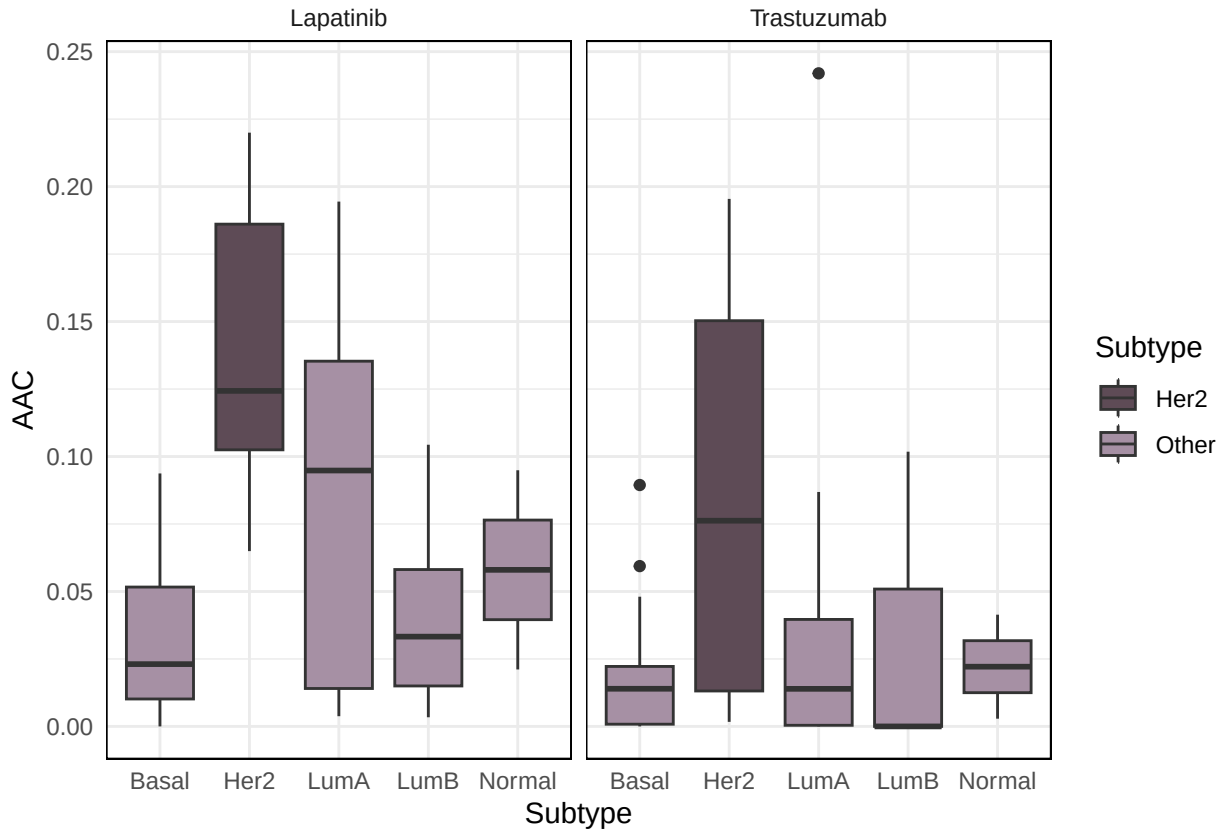
```
##        Subtype     Basal      Her2      LumA      LumB    Normal
## 184A1    Basal 0.7365824 0.0000000 0.0000000 0.0000000 0.2634176
## 184B5   Normal 0.2359258 0.0000000 0.1468017 0.0000000 0.6172726
## 21MT-1    Her2 0.2914219 0.7085781 0.0000000 0.0000000 0.0000000
## 21MT-2    Her2 0.0000000 0.8524511 0.0000000 0.1475489 0.0000000
## 21NT      Her2 0.3817810 0.6182190 0.0000000 0.0000000 0.0000000
## 21PT      Her2 0.3871703 0.6128297 0.0000000 0.0000000 0.0000000
```

Task: assess if response to HER2-targeted therapies is more prevalent in HER2 subtypes

```
# get drug sensitivity matrix
gray_sen <- summarizeSensitivityProfiles(
    gray,
    sensitivity.measure = "aac_recomputed",
    summary.stat = "median",
    verbose = TRUE
) |> t() |> as.data.frame()

# add drugs of interest
toPlot <- cbind(gray_pam50, gray_sen[,c("Trastuzumab", "Lapatinib")])
her2_toPlot <- toPlot %>%
  dplyr::select(Subtype, Trastuzumab, Lapatinib) %>%
  pivot_longer(
```

```
    cols = c(Trastuzumab, Lapatinib),
    names_to = "Drug",
    values_to = "AAC"
  )
her2_toPlot$label <- ifelse(her2_toPlot$Subtype == "Her2", "Her2", "Other")

ggplot(her2_toPlot, aes(x = Subtype, y = AAC, fill = label)) +
  geom_boxplot() + facet_wrap(~Drug) +
  theme_minimal() + theme(panel.border = element_rect()) +
  scale_fill_manual("Subtype", values = c("#5E4B56", "#A690A4"))
```

```
## Warning: Removed 62 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



Case scenario 2:

High tumour cellularity can be associated with more aggressive cancers. Gene signatures that are surrogate markers of tumour cellularity may be predictive of response to immunotherapies.

The code retrieves a dataset from orcestra.ca. The dataset is stored as a `PharmacoSet` object from the PharmacoGx package. The dataset contains 788 samples 675 cell lines with mutation and transcriptomic profiles, along with 16,688 drug response experiments.

```
gcsi <- readRDS(here("./data/rawdata/gCSI.rds")) |> updateObject()
gcsi
```

```
## <PharmacoSet>
## Name: gCSI
## Date Created: Sun May 24 23:50:46 2020
## Number of samples:  747
## Molecular profiles:
## Kallisto_0.46.1.rnaseq :
##   Dim: 60662, 675
## Kallisto_0.46.1.rnaseq.counts :
##   Dim: 60662, 675
## Kallisto_0.46.1.isoforms :
##   Dim: 227912, 675
## Kallisto_0.46.1.isoforms.counts :
##   Dim: 227912, 675
## CNV :
##   Dim: 26291, 329
## mutation :
##   Dim: 55, 329
## Treatment response: Drug pertubation:
##    Please look at pertNumber(cSet) to determine number of experiments  for each drug-sample combinati
## Drug sensitivity:
##    Number of Experiments:  6471
##    Please look at sensNumber(cSet) to determine number of  experiments for each drug-sample combinati
```

. . .

Task: extract the gene expression and mutation profiles

```
gcsi_rna <- summarizeMolecularProfiles(
  gcsi,
  mDataType = "Kallisto_0.46.1.rnaseq"
) |> suppressMessages()
```

```
##   |                                                                      |
```

```
gcsi_mut <- summarizeMolecularProfiles(
  gcsi,
  mDataType = "mutation",
  summary.stat = "and"
) |> suppressMessages()
```

Task: get treatment response data

```
gcsi_sen <- summarizeSensitivityProfiles(
    gcsi,
    sensitivity.measure = "aac_recomputed",
    summary.stat = "median",
    verbose = TRUE
)
```

Task: Find samples with lapatinib

```
egfr_tki <- c("Erlotinib", "Lapatinib")
genes <- c("EGFR", "BRCA1", "BRCA2")

# subset drug sensitivity
sen <- t(gcsi_sen[egfr_tki,]) |> as.data.frame()

# subset mutations
mut <- t(assay(gcsi_mut)) |> as.data.frame()
mut <- mut[,genes]

# subset gene expression
rna <- assay(gcsi_rna) |> as.data.frame()
keep <- gcsi_rna@elementMetadata$gene_id[gcsi_rna@elementMetadata$gene_name %in% genes]
rna <- t(rna[keep,]) |> as.data.frame()
colnames(rna) <- gcsi_rna@elementMetadata$gene_name[match(colnames(rna), gcsi_rna@elementMetadata$gene_

# keep common cell lines
common <- intersect(rownames(sen), intersect(rownames(rna), rownames(mut)))
sen <- sen[match(common, rownames(sen)),]
rna <- rna[match(common, rownames(rna)),]
mut <- mut[match(common, rownames(mut)),]
```

Task: is there an association between EGFR expression and TKi response

```
# plot associations between genes of interest and TKi
toPlot <- data.frame(
  sample = common,
  EGFR = rna[["EGFR"]],
  Lapatinib = sen[["Lapatinib"]],
  Erlotinib = sen[["Erlotinib"]]
)

# plot associations between genes of interest and TKi
plot_egfr <- function(drug) {
  return(
    ggplot(toPlot, aes(x = EGFR, y = .data[[drug]])) +
      geom_point() + geom_smooth(method = "lm") +
      theme_minimal() +
      labs(x = "EGFR expression (TPM)", y = paste(drug, "response (AAC)"), title = drug)
  )
}

p1 <- plot_egfr("Lapatinib")
p2 <- plot_egfr("Erlotinib")

ggarrange(p1, p2)
```

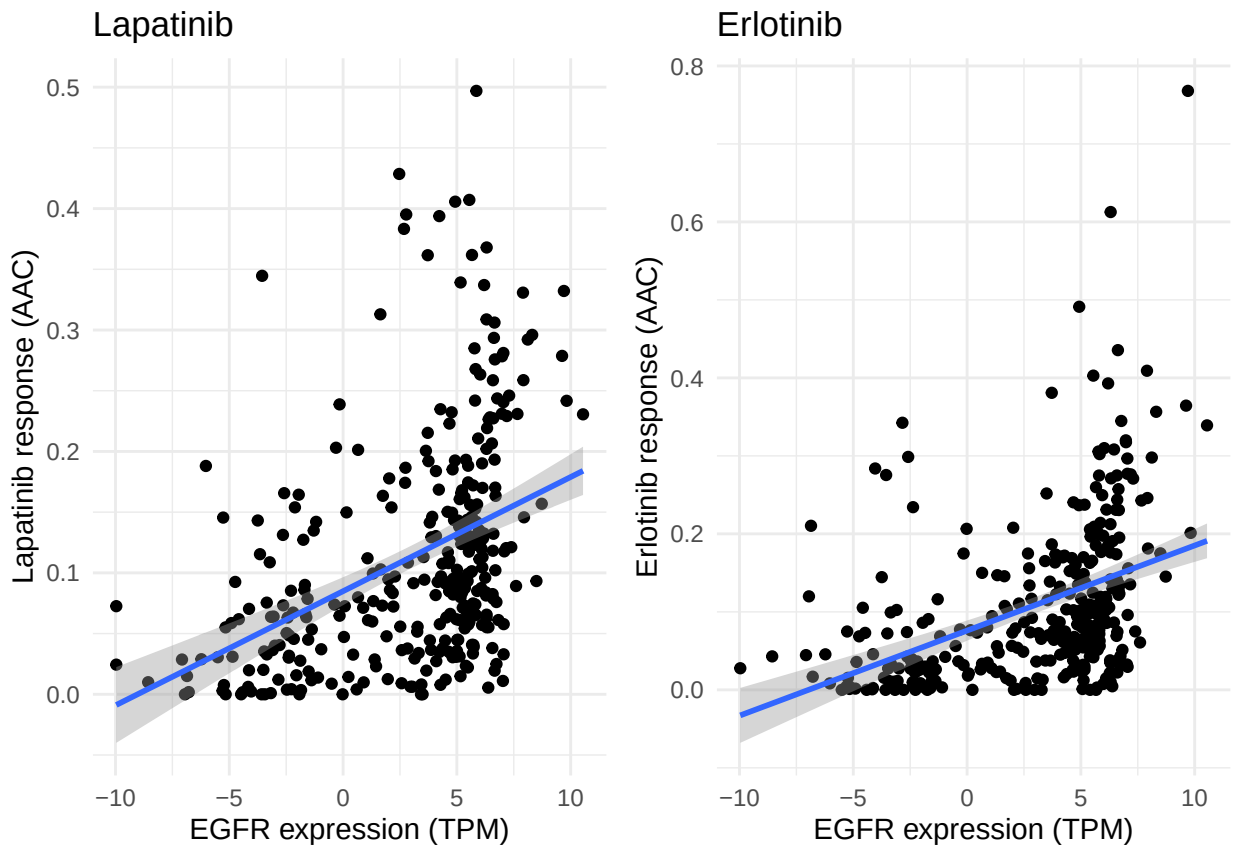
```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 416 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 416 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 415 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 415 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Task: does BRCA mutation status influence the strength of association

```
# get samples with BRCA1 mutation
mutated <- mut[!is.na(mut$BRCA1),] |> rownames()
toPlot$brca1 <- ifelse(toPlot$sample %in% mutated, "Mutated", "Unmutated")

# plot association faceted by BRCA1 mutation status
plot_egfr <- function(drug) {
  return(
    ggplot(toPlot, aes(x = EGFR, y = .data[[drug]])) +
      geom_point() + geom_smooth(method = "lm") +
      facet_wrap(~brca1, ncol = 1) +
      theme_minimal() +
      labs(x = "EGFR expression (TPM)", y = paste(drug, "response (AAC)"), title = drug)
```

```
  )
}

p1 <- plot_egfr("Lapatinib")
p2 <- plot_egfr("Erlotinib")
ggarrange(p1, p2)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 416 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 416 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 415 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 415 rows containing missing values or values outside the scale range
## (`geom_point()`).
```