

DRUMLR Tutorial

Henry Gerdes

10/09/2020

Contents

0.1	Set up	2
1	Building DRUML models with DRUMLR	3
1.1	Generate markers	3
1.2	Build Models	3
1.3	Verification with External Data	3
2	Verification of Gerdes et al DRUML models using clinical AML data	4
2.1	Import Internal Data	4
2.2	Calculate Distance values	4
2.3	AAC prediction using Gerdes et al models	4
2.4	Rearrange Prediction output for graphs	5
2.5	Verification graphs	6

0.1 Set up

To install the DRUMLR package into R ensure you have installed the devtools package and run the following code.

```
devtools::install_github(repo="CutillasLab/DRUMLR", subdir="DRUMLR_code")
```

1 Building DRUML models with DRUMLR

Load Libraries

```
library(tidyverse)
library(DRUMLR)
```

1.1 Generate markers

Import and scale input data. Then use the processed data to build markers for DRUMLR models.

```
#import datasets
df_input <- Input_data %>% scale(center = T, scale = T)
df_response <- Response_data

#Generate EMDRs
markers <-
  DRUMLR::BuildEMDR(
    df_input = Input_data,
    df_response = t(df_response),
    drugs = drugs,
    computational_load = 0.8
  )
```

1.2 Build Models

Use the BuildDRUMLs function to build DRUML models. The internal function .BuildModels contains the parameters used for model tuning

```
Model_info <-
  DRUMLR::BuildDRUMLs(
    df_input = df_input_aml,
    .marker_database = markers,
    input_type = "aml_phospho",
    save_path = "~/DRUML_models",
    save_csv = T,
    models = "all",
    df_response = df_response,
    drugs = drugs,
    computational_load = 0.8
  )
```

1.3 Verification with External Data

Prediction of verification data shown in verification data example.

```
df_validation <- external_data

df_validation_dist <-
  DRUMLR::DrugMarkerEnrichment(df = df_validation, marker_database = markers)

df_predictions <-
  DRUMLR::PredictDRUML(df_distance = df_validation_dist,
    models = "all",
    models_dir = "~/DRUML_models")
```

2 Verification of Gerdes et al DRUML models using clinical AML data

Load Libraries

```
library(DRUMLR)
library(tidyverse)
library(survival)
```

2.1 Import Internal Data

Import internal DRUML data for verification.

```
#Import and scale Verification Phosphoproteomic input data from clinical aml samples
df_input <- DRUMLR:::Primary_AML_phospho_data

#Import survival and molecular classification data for clinical AML samples
df_survival <- DRUMLR:::Primary_AML_survival_data
```

2.2 Calculate Distance values

In this example we will predict the sensitivity of clinical_phosphoproteomics samples to Cytarabine. To generate rankings peptide abundance values are scaled for each sample. Distance values for these data are generated using the DRUMLR::DrugMarkerEnrichment() function.

The markers_databases which can be used in for this are:

- Phosphoproteomics:
 - DRUMLR:::phospho_aml_markers.
 - DRUMLR:::phospho_solid_markers.
- Proteomic:
 - DRUMLR:::prot_aml_markers.
 - DRUMLR:::prot_solid_markers.

In this case we will be using phospho_aml_markers

```
# Get relevant marker database. For this example we are using phosphoproteomics
# data
markers <- DRUMLR:::phospho_aml_markers

# Carry out Drug Marker Enrichment on the data using the relevant markers database
df_distance <- DRUMLR::DrugMarkerEnrichment(df = df_input, marker_database = markers)
```

2.3 AAC prediction using Gerdes et al models

To predict Drug Sensitivity you will need to state the path to the folder containing all of the Example Models. In this example we are predicting Cytarabine sensitivity using partial list squares (pls), principal components regression (pcr) and random forest (rf) models.

```
#To predict Drug Sensitivity you will need to state the path to the folder containing all of the Models

predictions <- DRUMLR::PredictDRUML_example(df_distance = df_distance,
      input_type = "phospho",
      cancer_type = "aml",
      models = c("rf","pcr","pls"),
      drugs = "Cytarabine",
```

```
models_dir = "Example_Models"  
)
```

2.4 Rearrange Prediction output for graphs

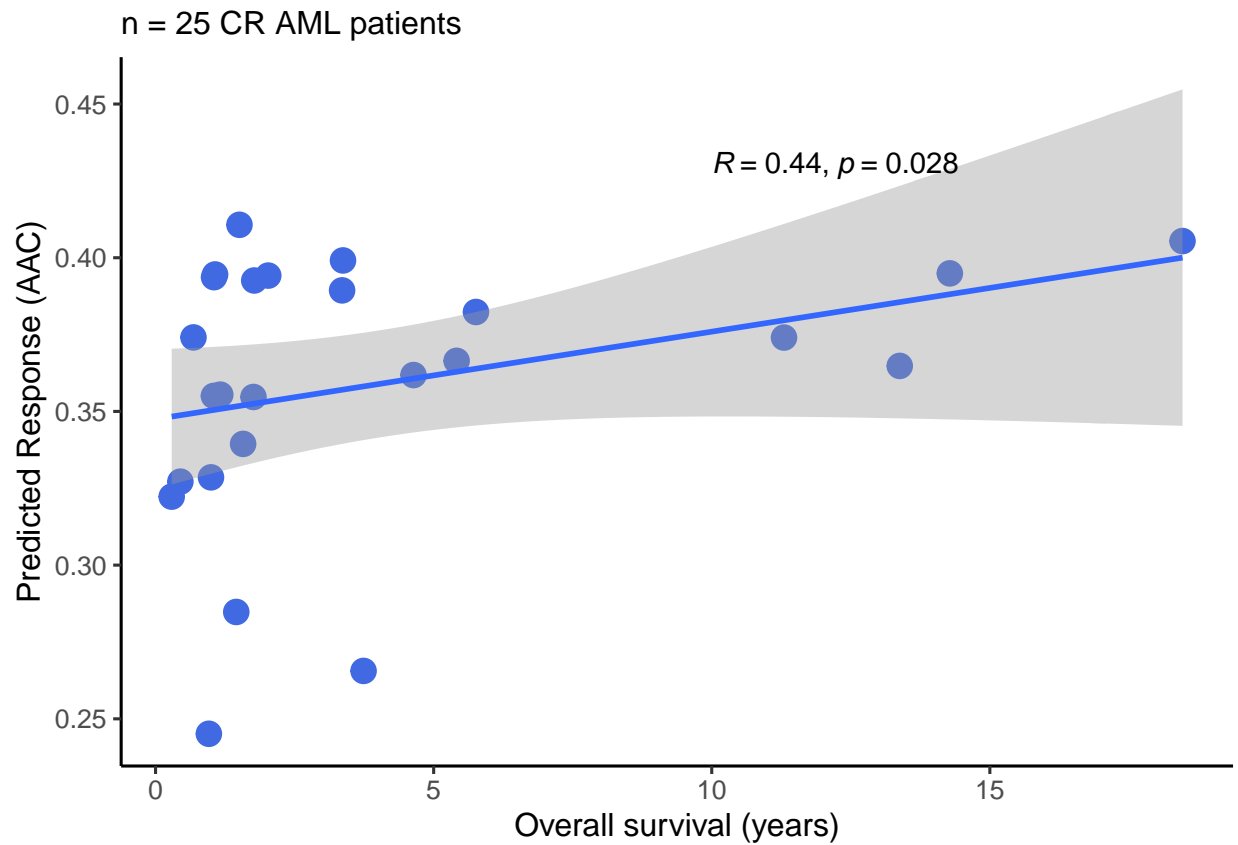
Useful code for reshaping prediction results for ggplot2 based graphs.

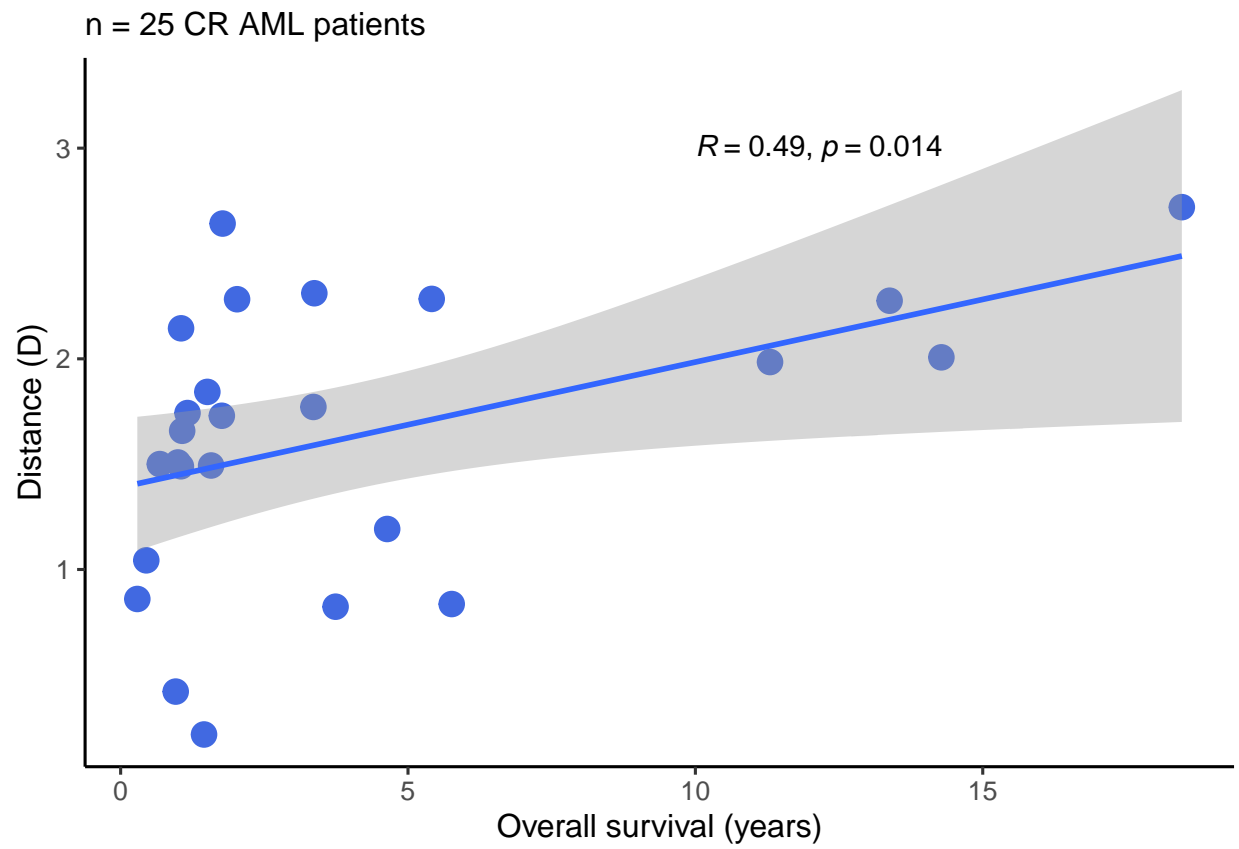
```
# Get average of predictions  
df_prediction2 <- t(predictions) %>% data.frame()  
df_prediction2["Cytarabine_aml_median", ] <- apply(df_prediction2, MARGIN = 2, median)  
# label Predictions outputs  
df_labs <- strsplit(rownames(df_prediction2), "_") %>% data.frame() %>% t()  
colnames(df_labs) <- c("drug", "tissue", "model")  
  
# labeled ML predictions  
df_prediction2 <- cbind(df_labs[, "model"], df_prediction2) %>% reshape2::melt()  
colnames(df_prediction2) <- c("model", "Vial.ID", "aac")  
  
# Add survival and D data to dataframe  
df_prediction2 <- merge.data.frame(df_prediction2, df_survival, by = "Vial.ID")  
df_prediction2$D <- df_distance["Cytarabine", df_prediction2$Vial.ID] %>% as.vector()
```

2.5 Verification graphs

2.5.1 Correlation of Predicted AAC with Patient Survival

Scatter graphs demonstrating the correlation between median predicted AAC values against patient survival rate in years.

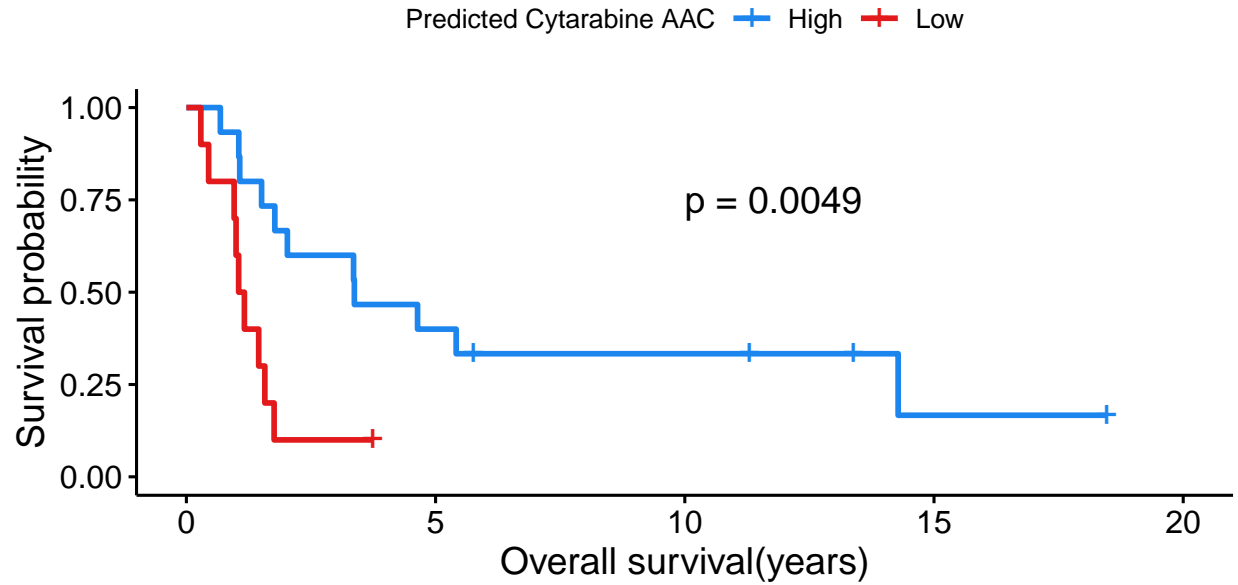




2.5.2 Make Kaplan Meyer classifications

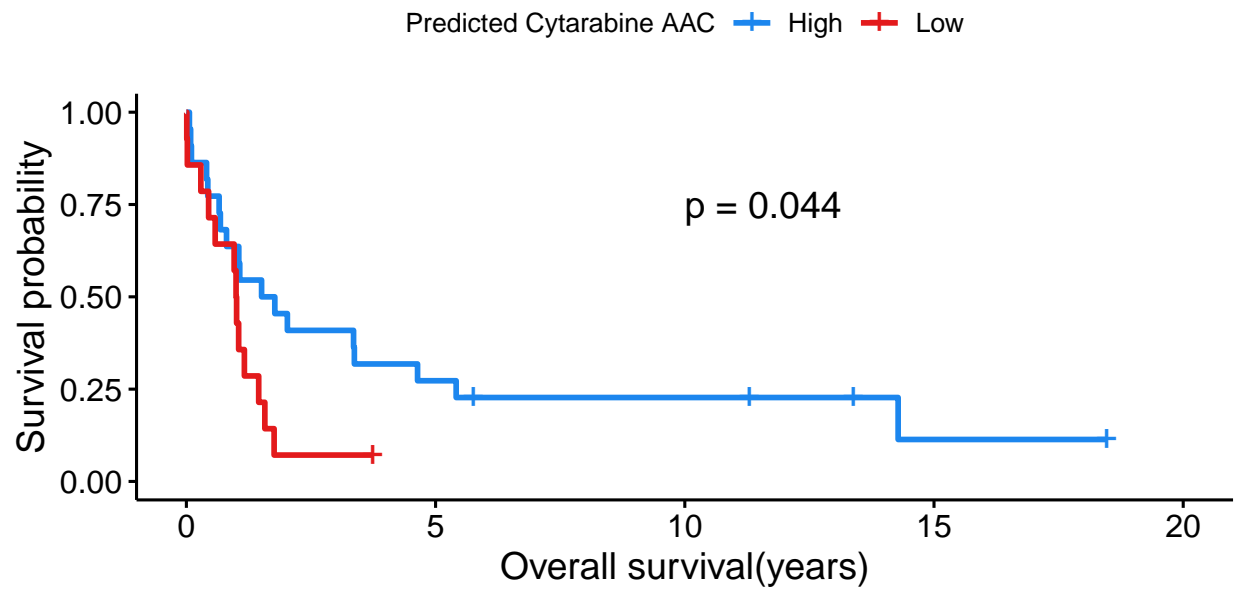
Patients were split into “high” and “low” groups using mean AAC and D values.

Patient OS, CR patients only



	records	*rmean	median	0.95LCL	0.95UCL
high	15	7.12	3.37	1.77	NA
low	10	2.82	1.11	0.96	NA

Patient OS, including refractory



	records	*rmean	median	0.95LCL	0.95UCL
<i>high</i>	22	4.97	1.64	0.8	5.41
<i>low</i>	14	2.13	1	0.58	1.76