
GenomicLearning: Learning gene expression with regression

Running R studio cloud

Please make an Rstudio account and login here: <https://rstudio.cloud/>

Also you can download and install Rstudio on your own computer: <https://www.rstudio.com>

We will work with the console window in Rstudio for this exercise with logistic regression (described below)

Installing R package GenomicLearning from github

The R package GenomicLearning is freely available to download and distribute from github <https://github.com/radamsRHA/GenomicLearning/>. To install and load GenomicLearning, you must first install the R package devtools,

```
install.packages("devtools")
```

Now using devtools we can install GenomicLearning from github:

```
library(devtools)
install_github("radamsRHA/GenomicLearning")
library(GenomicLearning) # Load package
```

to follow along with this tutorial, GenomicLearning also requires the following dependencies to be installed:

```
install.packages("devtools")
install.packages("glmnet")
install.packages("ggplot2")
```

```
library(devtools)
library(GenomicLearning)
library(ggplot2)
```

Tutorial with logistic regression

Step 1) load input packages

```
library(devtools)
install_github("radamsRHA/GenomicLearning")
library(GenomicLearning)
library(ggplot2)
library(dplyr)
```

Step 2) visualize SingleCellExperiment

```
head(SingleCellExperiment)
```

Step 4) always visualize your data!

```
plot(VenomPresent ~ TF01, data = SingleCellExperiment)
boxplot(TF01 ~ VenomPresent, data = SingleCellExperiment)
```

Step 5) Visualize logistic regression model

```
SingleCellExperiment %>%
  ggplot(mapping = aes(x = TF01, y = VenomPresent)) + geom_point() +
  geom_smooth(method = "glm",
              method.args = c(family = "binomial"))
```

Step 6) Visualize linear regression model

```
SingleCellExperiment %>%
  ggplot(mapping = aes(x = TF01, y = VenomPresent)) +
  geom_point() +
  geom_smooth(method = "glm",
              method.args = c(family = "gaussian"))
```

Step 7) Fit GLM model for logistic regression

```
glm.fit <- glm(VenomPresent ~ TF01, SingleCellExperiment, family = "binomial")
summary(glm.fit)
```

Step 9) Make a prediction for a cell with 25 TF01 counts

```
predict(glm.fit, tibble(TF01 = c(35)), type = "response")
```

Step 10) Compute probabilities on training data

```
glm.probs <- predict(glm.fit, type = "response")
```

Step 11) Append computed probabilities to a new data frame

```
SingleCellExperiment.withProbs <- SingleCellExperiment %>%
  mutate(probs = glm.probs,
         pred = ifelse(probs > 0.5, 1, 0))
```

Step 12) Compute confusion matrix

```
SingleCellExperiment.withProbs %>% select(VenomPresent, pred) %>% table()
```