R

Tablular Data

Reading tables:

```
df = read.table("path/filename.csv") # default tab separated
df = read.csv("path/filename.csv") # default comma separated
df = data.table::fread("path/filename.csv", data.table = F) # fast -- preferred!
```

Writing tables:

```
write.csv(df, "path/filename.csv") # default comma separated
data.table::fwrite(df, "path/filename.csv") # fast -- preferred!
```

Remember that we examined three kinds of data in TCGA. Data stored as text (`.csv`) in **bold**

1. **Clinical**.
2. Transcriptomic.
3. **Mutation**.

SummarizedExperiments (Transcriptomics)

Our transcriptomic data was stored as a `SummarizedExperiment`. Before, we queried it every time, but there's a better way to store it using the `HDF5Array` package. It'll actually create a separate folder.

```
# installation -- do this once
# BiocManager::install("HDF5Array")
library(HDF5Array)

# reading
HDF5Array::HDF5Array("path/filename.h5", name="sum_exp")

# writing
HDF5Array::writeHDF5Array(sum_exp,
                          "path/analysis_data/filename.h5",
                          name="sum_exp",
                          with.dimnames = T)
```

MAFs (Mutations)

We stored our MAF data as a giant csv. To convert the csv into a MAF object, we did the following:

```
library(maftools)
maf_dataframe = data.table::fread("path", data.table = F)
clinic = data.table::fread("path", data.table = F)
maf_object = read.maf(maf = maf_dataframe,
                      clinicalData = clinic,
                      isTCGA = TRUE)
```

Figures

We didn't really discuss how to save figures in R because the syntax is a bit weird, but here is an example below:

```
png("path/img.png")
# code to generate your plot
dev.off()
```

Python

Tabular Data

We'll use `pandas` to open and write tables:

```
import pandas as pd

df = pd.read_csv("path/file.csv")
# do something with df
df.to_csv("path/file.csv")
```

Figures

Instead of using `plt.show()`, we just use `plt.savefig()`:

```python
import matplotlib.pyplot as plt

fig, ax = plt.subplots(1, 1)
# make your plot
plt.savefig("path/img.png")
```