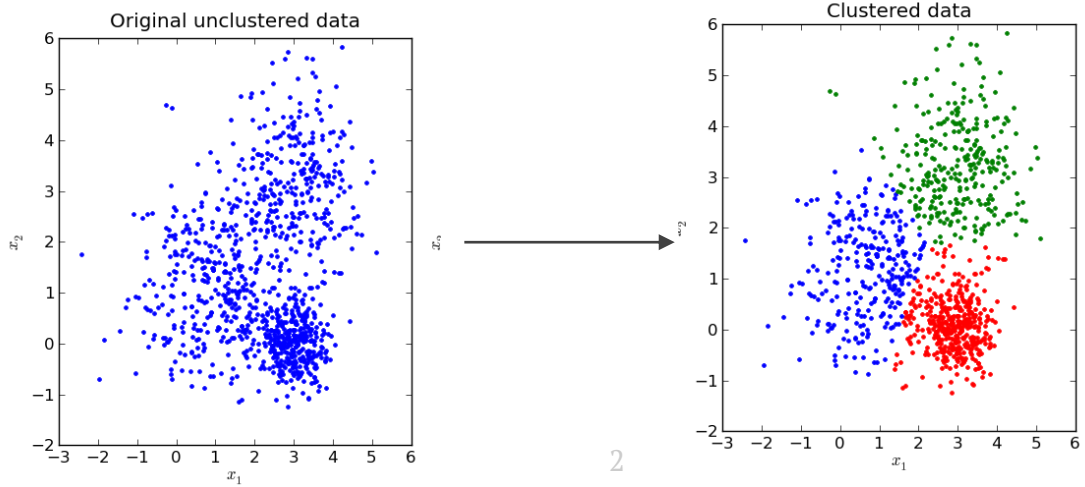
Out of Tune
Reproducible Machine Learning

Statistics and Machine Learning

- S - Statistical
- L - Learning
- D - Data
- M -Mining

Combining statistics, mathematics, and computing in order to analyze complex sources of data



Machine Learning is Ubiquitous

- Machine learning is a powerful tool to solve important bioinformatics problems.

Early triage of critically ill COVID-19 patients using deep learning

Wenhua Liang, Jianhua Yao, [...] Jianxing He

Diagnostic evaluation of a deep learning model for optical diagnosis of colorectal cancer

Dejun Zhou, Fei Tian, Xiangdong Tian, Lin Sun, Xianghui Huang, Feng Zhao, Nan Zhou, Zuoyu Chen, Qiang Zhang, Meng Yang, Yichen Yang, Xuexi Guo, Zhibin Li, Jia Liu, Jiefu Wang, Junfeng Wang, Bangmao Wang, Guoliang Zhang, Baocun Sun, Wei Zhang, Dalu Kong, Kexin Chen & Xiangchun Li

Machine learning applied to enzyme turnover numbers reveals protein structural correlates and improves metabolic models

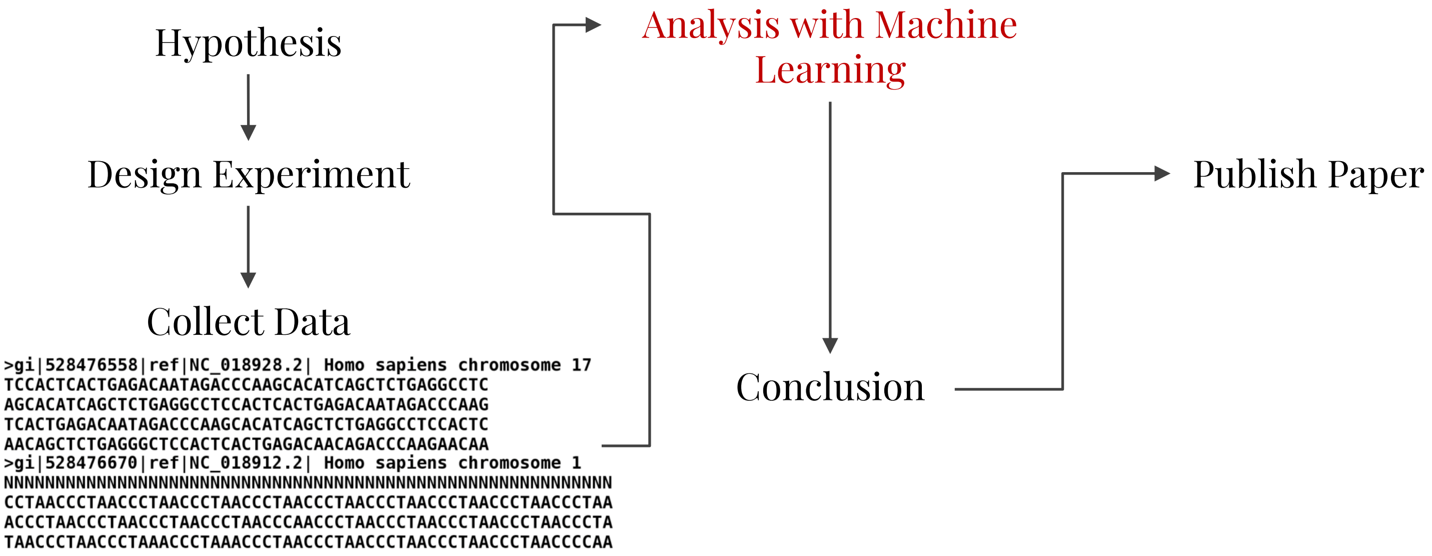
David Heckmann, Colton J. Lloyd, Nathan Mih, Yuanchi Ha, Daniel C. Zielinski, Zachary B. Haiman, Abdelmoneim Amer Desouki, Martin J. Lercher & Bernhard O. Palsson

Application of combinatorial optimization strategies in synthetic biology

Gita Naseri & Mattheos A. G. Koffas

Machine Learning is Ubiquitous

- We often rely on machine learning techniques in our bioinformatics research.



Reanalyzing Data

- Often we need to run an algorithm from a published paper

The Dreaded Sentence

"We conducted our analysis using the *FancyStatistics* Package."

“We used a Package”

The entire path of solutions (in λ) for the ridge regression, lasso and elastic net models were computed using the pathwise cyclical coordinate descent algorithms-- computationally efficient methods for solving these convex optimization problems-- in *glmnet* in R [12].

adaptive lasso was fit using the *parcor* package in R

adaptive elastic net using an R function that calls the *elasticnet*

This section describes some packages for genetic data analysis according to their package descriptions in CRAN. They fall into several categories: data manipulation (*genetics*); phylogenetic analysis (*PHYLOGR, ape*); association analysis of population data including population structure (*biodem, genetics, hapassoc, haplo.score, haplo.stats, hierfstat, hwde, ldDesign, LDheatmap, Malmig, popgen, R/gap, rmetasim*); family data (*tdthap*); and QTL for experimental design (*bim, bqtl, happy, qtlDesign, R/qtl*). Others (*BradleyTerry, epitools, evd, gllm, locfdr, rmeta, vcd*) are fairly general and are not limited to analysis of genetic data. There are a large number of packages for microarray analysis, as described below.

Is My Analysis Reproducible?

"The LASSO regression was run via the *glmnet* R package."

Is My Analysis Reproducible?

Saying the package alone may not be enough.

Let's take a look at the *help* page for *glmnet*

Is My Analysis Reproducible?

Usage

```
glmnet(x, y, family = c("gaussian", "binomial", "poisson", "multinomial",
  "cox", "mgaussian"), weights, offset = NULL, alpha = 1,
  nlambda = 100, lambda.min.ratio = ifelse(nobs < nvars, 0.01, 1e-04),
  lambda = NULL, standardize = TRUE, intercept = TRUE,
  thresh = 1e-07, dfmax = nvars + 1, pmax = min(dfmax * 2 + 20,
  nvars), exclude, penalty.factor = rep(1, nvars), lower.limits = -Inf,
  upper.limits = Inf, maxit = 1e+05, type.gaussian = ifelse(nvars <
  500, "covariance", "naive"), type.logistic = c("Newton",
  "modified.Newton"), standardize.response = FALSE,
  type.multinomial = c("ungrouped", "grouped"), relax = FALSE,
  trace.it = 0, ...)
```

Is My Analysis Reproducible?

Usage

```
glmnet(x, y, family = c("gaussian", "binomial", "poisson", "multinomial",
  "cox", "mgaussian"), weights, offset = NULL, alpha = 1,
  nlambda = 100, lambda.min.ratio = ifelse(nobs < nvars, 0.01, 1e-04),
  lambda = NULL, standardize = TRUE, intercept = TRUE,
  thresh = 1e-07, dfmax = nvars + 1, pmax = min(dfmax * 2 + 20,
  nvars), exclude, penalty.factor = rep(1, nvars), lower.limits = -Inf,
  upper.limits = Inf, maxit = 1e+05, type.gaussian = ifelse(nvars <
  500, "covariance", "naive"), type.logistic = c("Newton",
  "modified.Newton"), standardize.response = FALSE,
  type.multinomial = c("ungrouped", "grouped"), relax = FALSE,
  trace.it = 0, ...)
```

There are 20+ options in this one function!

Tuning Parameters

Tuning parameters are options that must be determined by the scientist to fit a *given* model to the data.

Tuning Parameters

Tuning parameters include things such as:

1. The number of clusters in an algorithm.
2. The starting points for algorithms.
3. The number of iterations an algorithm runs for.
4. The strength of any penalties in the algorithm.
5. Many, many, *many* more

Tuning Parameters

The choice of tuning parameters can drastically affect the outcome of your analysis.

Example

A food science researcher wants to predict the intake of different foods on an individual's life expectancy.

"We used the *glmnet* package in R."

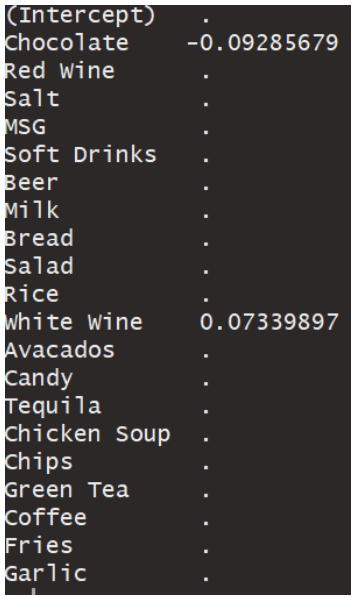
```
(Intercept)    .
Chocolate     -0.10649903
Red Wine       .
Salt           .
MSG            .
Soft Drinks    .
Beer           .
Milk           .
Bread          .
Salad          .
Rice           .
White Wine     0.08848753
Avacados       .
Candy          .
Tequila       -0.01873881
Chicken Soup   .
Chips          .
Green Tea      .
Coffee         .
Fries          .
Garlic         .
```

"Chocolate and tequila lowers life expectancy and white wine increases life expectancy."

Example

A food science researcher in a major tequila producing state reanalyzes the data the data from the first study.

"We used the *glmnet* package in R."



"Actually, tequila doesn't have any negative effect on life expectancy."

What's the Difference?

```
set.seed(01)
cvAll = cv.glmnet(x,y, intercept = F)
set.seed(01)
cvTequila = cv.glmnet(x,y, intercept = F, nlambda = 25)
```

Both researchers analyzed the same data, used the same function, but the researcher with an investment in tequila **changed the nlambda tuning parameter from the default nlambda = 100 to nlambda = 25.**

Whose Analysis was Correct?

Technically the first analysis but we wouldn't know that without the code.

1. Neither individual told us what the tuning parameters were.
2. Neither individual had any reasoning to why they chose the parameters to what they wee.

What Can We Do?

- In the main text, have an ***easily*** understandable explanation of your analysis algorithm.
- Publish code used in analysis or simulations (github/CRAN/Supplementary Material), with information on **software and package versions**.
- Use a clear README or help file when explaining simulations
- Clearly explain why certain choices in algorithm design were taken.
- **Learn why the default is the default** (sometimes it's completely arbitrary!)

Thank you!