Recent developments in the design and analysis of clinical trials.

Javier Cabrera

Dept of Statistics, Dept of Medicine, Rutgers University

Outline

- Part I: Analysis of Biased and poorly randomized clinical studies
- Part II Alternative study designs.
- Part III. Machine learning and deep learning methods for data augmentation and matching populations.

Part I

Analysis of Biased and poorly randomized clinical studies

- Introduction. Poorly randomized and biased clinical studies.
- Super Learners for nonlinear function estimation.
- Propensity scores estimation using Super Learners and other methods.
- TMLE procedure

Introduction

In this lectures we will introduce the following topics:

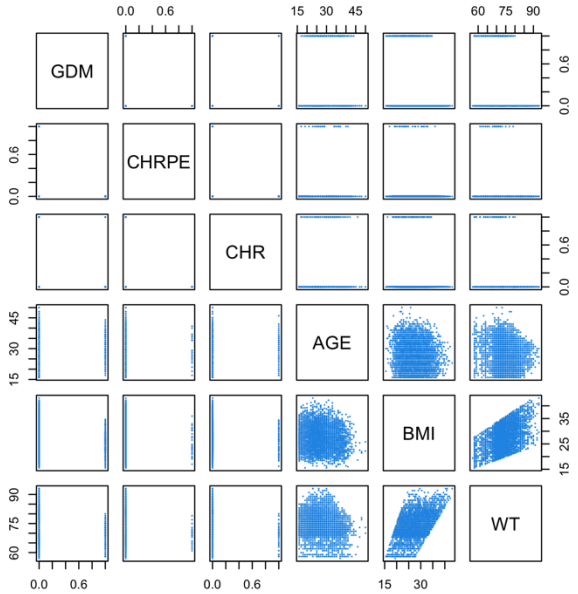
- Poorly randomized and biased clinical studies.
- SuperLearner. TMLE
- Indices: Propensity Scores, Differential Natural Hermite.
- Genetic Algorithm(GA)
- Augmenting Controls, Animal Studies, Rebalancing biased studies
- Datanuggets and Large Datasets/Big Data Matching
- Deep Learning, Auto Encoders/Decoders, Variational auto-encoder
- Expanding Applications to Large Datasets/Big Data.

A placenta abruption study

A study on preventing placenta abruption on women in NJ (PACER,2023) . Want to use real world controls from pregnancy database of New Jersey hospital births. We have 18 variables (25 features) in common between the clinical study and the real-world database. (Limitation)

We use synthetic data: *copydata* function on DNAMR

Scatter matrix of 6 of the 25 features in dataset.



Variables

MONTH
PE_MILD
PE_SEVERE
REGION
RACE
PRE_DM
OLIGO
MARITAL
MULTIPLE
HOSPBEDR
HOSPOWN
GES_HYP
GDM
CHRPE
CHR
AGE
BMI
WT

A placenta abruption study

- Angioedema disease of swelling of tissue all over the body. In some cases angioedema can be fatal.
- ACE inhibitors were discovered in Brazil from the poison of an Amazonian snake.
- Used for treating hypertension.
- Clinical study: ACE inhibitors may increase dead by angioedema patients?
- Outcome: Dead at 5 years follow up.
- Data: “acetrial.csv”.

Variables

The variables are:

Dead at 5 years:	0 censored, 1 dead
treat:	0 control, 1 ACE treated
age:	
drink, smoke,diab:	0 No, 1 Yes
gender:	0 Male, 1 Female
race:	0 White, 1 Black, 2 Other
sbp:	Systolic Blood Pressure.

Poorly randomized and biased clinical studies.

- Poorly randomized and biased clinical studies.
- Classical analysis: Linear models
  - A: Treatment
  - W: Confounders => model Misspecified
  - No interactions
- Model estimation and inference (A-priori)
  - Try many models and report the best: Bias and overfitting
  - Treatment effect may pickup model missing interactions

Causal Inference: Estiment = Average Treatment Effect (ATE)
(Compared to Estimator, Estimate)

Statistics => Data Analysis => Data Mining => AI/ML

AI/ML + Stats => Good predictive models => SuperLearner

Super Learners for nonlinear function estimation

SuperLearner algorithm:

- Combines ML methods: GLM :: LASSO/ENET :: Random Forest :: GAM :: SVM :: NNET ::…
- Uses cross-validation creates an optimal weighted average of those models, aka an "ensemble", using the test data performance. Possible to choose fold : 5, 10,20, leave-one out CV

Option: **SuperLearner.CV.control(V = 5L)**

- It is asymptotically the best possible prediction algorithm that has been tested.
- It is computationally intensive

```
library(SuperLearner)
sl_libs <- c('SL.glmnet', 'SL.gam', 'SL.earth','SL.nnet','SL.glm')
Q <- SuperLearner(Y=d[,1,X=d[,-1],family=binomial(),SL.library=sl_libs,
                                  cvControl=SuperLearner.CV.control(V = 5L) )
PE=  predict(Q)$pred
```

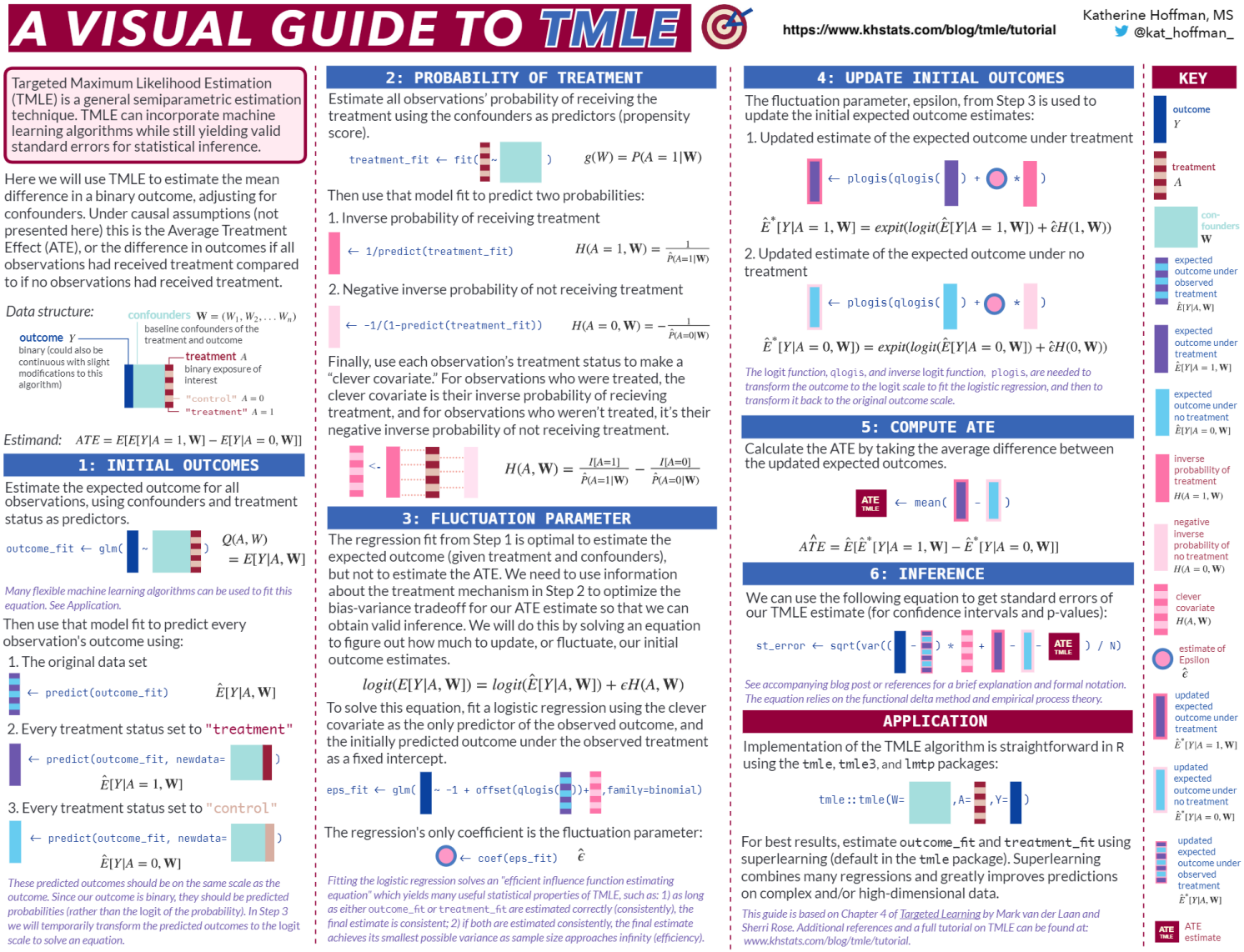
Propensity scores estimation using Super Learners

Propensity Score Functions

- Treatment assignment variable: A { 1 Treatment , 0 Control }
- Confounders: $W = \{W_1, \ldots, W_{p-1}\}$
- PE model: $PE = P(A=1 \mid W) = E(A \mid W)$.
- This was estimated using logistic model, or ML methods, RF, SVM, gam or other
- Use SuperLearner To estimate PE, choose a set of methods that you prefer. Generally use Lasso :: Random Forest :: GAM :: Mars :: NNET

TMLE

https://www.khstats.com/blog/tmle/tutorial



TMLE

Step 1: Estimate the Outcome

$$Q(A, \mathbf{W}) = \mathrm{E}[Y|A, \mathbf{W}]$$

Q_A: all observations

Q_1: If every observation received the treatment.

$$\hat{\mathrm{E}}[Y|A = 1, \mathbf{W}]$$

Q_0: If every observation received the control.

$$\hat{\mathrm{E}}[Y|A = 0, \mathbf{W}]$$

Causal Inference: Estimand = Average Treatment Effect

(Compare to Estimator, Estimate)

$$\hat{ATE}_{G-comp} = \hat{\Psi}_{G-comp} = \frac{1}{N}\sum_{i=1}^{N}(\hat{\mathrm{E}}[Y|A = 1, \mathbf{W}] - \hat{\mathrm{E}}[Y|A = 0, \mathbf{W}])$$

TMLE

Step 2: Estimate the Probability of Treatment

$$g(\mathbf{W}) = \Pr(A = 1|\mathbf{W})$$

The inverse probability of receiving treatment is:

$$H(1, \mathbf{W}) = \frac{1}{g(\mathbf{W})} = \frac{1}{\Pr(A = 1|\mathbf{W})}$$

The negative inverse probability of not receiving treatment is:

$$H(0, \mathbf{W}) = -\frac{1}{1 - g(\mathbf{W})} = -\frac{1}{\Pr(A = 0|\mathbf{W})}$$

TMLE

“Clever” Covariate:

$$H(A, \mathbf{W}) = \frac{\mathrm{I}(A = 1)}{\Pr(A = 1|\mathbf{W})} - \frac{\mathrm{I}(A = 0)}{\Pr(A = 0|\mathbf{W})}$$

Steps 3- 4: Update the Initial Estimates of the Expected Outcome

$$\hat{\mathrm{E}}^*[Y|A, \mathbf{W}] = expit(logit(\hat{\mathrm{E}}[Y|A, \mathbf{W}]) + \hat{\epsilon}H(A, \mathbf{W}))$$

$$\hat{\mathrm{E}}^*[Y|A = 1, \mathbf{W}] = expit(logit(\hat{\mathrm{E}}[Y|A = 1, \mathbf{W}]) + \hat{\epsilon}H(1, A))$$

$$\hat{\mathrm{E}}^*[Y|A = 0, \mathbf{W}] = expit(logit(\hat{\mathrm{E}}[Y|A = 0, \mathbf{W}]) + \hat{\epsilon}H(0, W)$$

TMLE

Step 5: Compute the Statistical Estimand of Interest

$$\hat{ATE}_{TMLE} = \hat{\Psi}_{TMLE} = \frac{1}{N}\sum_{i=1}^{N}[\hat{E}^*[Y|A=1,\mathbf{W}] - \hat{E}^*[Y|A=0,\mathbf{W}]]$$

Step 6: Calculate the Standard Errors for Confidence Intervals and P-values

$$\hat{IF} = (Y - \hat{E}^*[Y|A,\mathbf{W}])H(A,\mathbf{W}) + \hat{E}^*[Y|A=1,\mathbf{W}] - \hat{E}^*[Y|A=0,\mathbf{W}] - \hat{ATE}$$

$$\hat{SE} = \sqrt{\frac{var(\hat{IF})}{N}}$$

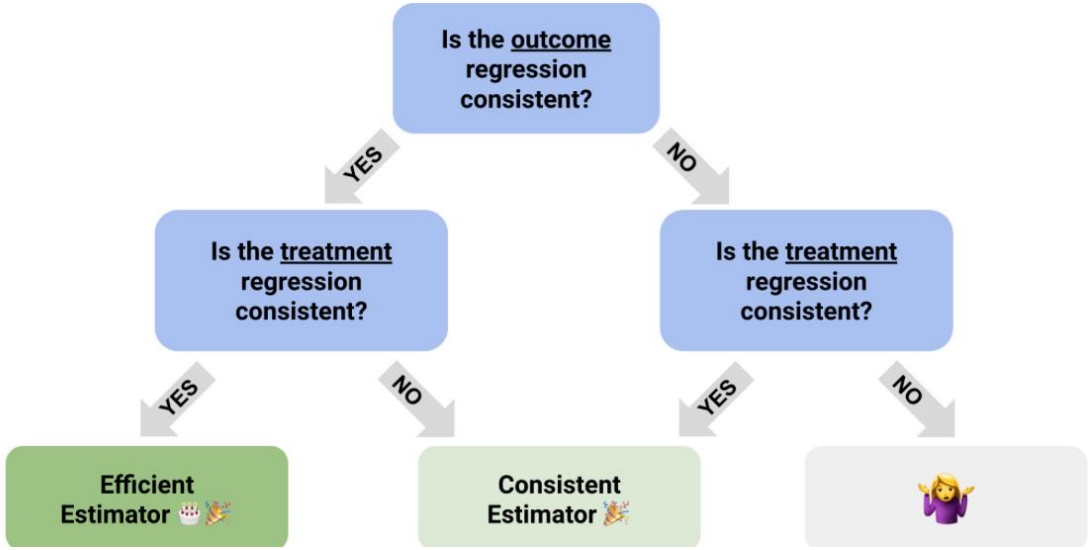
TMLE

Properties of TMLE

It allows the use of machine learning algorithms while still yielding asymptotic properties for inference.

TMLE is a **doubly robust** estimator: If both models for Y or for the probability of treatment are consistent then TMLE is consistent

If both regressions are consistent, the **final estimate will reach the smallest possible variance at a rate of root n**



TMLE

Practice:

Problem 1. Practice with the file “TMLE.Rmd”

Problem 2. Use the acetrial dataset in the file “acetrial.csv”
response = death
treatment = treat
Minimum covariates = Gender, Smoke, BMI and SBP

- Estimate the propensity scores function using SuperLearner
- Fit TLME model calculate estimand for ATE

Part II

Alternative study designs

- Introduction. Matching populations.
- Differential Natural Hermite Index (DNHI) to measure distances between populations.
- Propensity Scores Index (PSI), a simpler alternative DNHI.
- Genetic algorithm to match populations.
- Applications of GA, DNHI and PSI to Augmenting control groups from real world databases.
- Applications of GA, DNHI and PSI to randomization of animal studies.
- Algorithm for sequential randomization to rebalance clinical studies

BACKGROUND

- An alternative to TMLE is to improve the design of the study making sure that the resulting populations are matched in distribution.

- The following are examples where this is possible

  - Augmenting Clinical Data from partly(or fully) synthetic control arm from real world data or claims data. Reduces cost and allows for better trials for rare diseases.

  - Animal studies. The animals are available at the beginning of the study and need to be split into groups.

  - Improving randomization of a clinical study. At some interim analysis point it is found that the data is unbalanced (sometimes 25%). How to rebalance the study.

Matching populations.

- Matching or matched-pairs is a standard technique to estimate ATE, ATT, ATC.
  - ATC is the Average treatment effect on the treated and
  - ATC is the Average treatment effect on the controls
- One of the best know algorithms for matching is implemented in the R package Matchit that includes many types of matching: Optimal, Exact, Coarsed… and distances such as Euclidean, Propensity,…
- Another simpler way to match is to match with constrains:

- Some variables should be match exactly: Gender, Race,…
- Some should be matched with a max difference: Age, Year,…
- The rest, often comorbidities and other could be match propensity score distance of other distance.

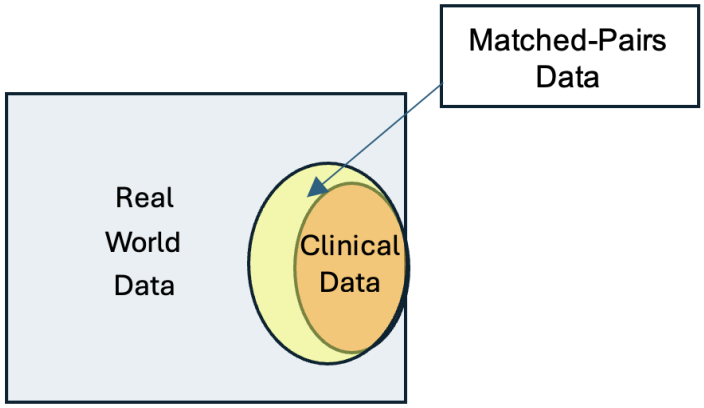
Some drawbacks of Matching.

- Matching often discards observations.
  Often not all observations are matched.

- When matching a smaller dataset (clinical study) with a larger dataset (real-world) there maybe a bias because clinical studies tend to use nearly healthy patients.

- For smaller studies matching could be too specific.

- For very large datasets Matching could be very expensive computationally

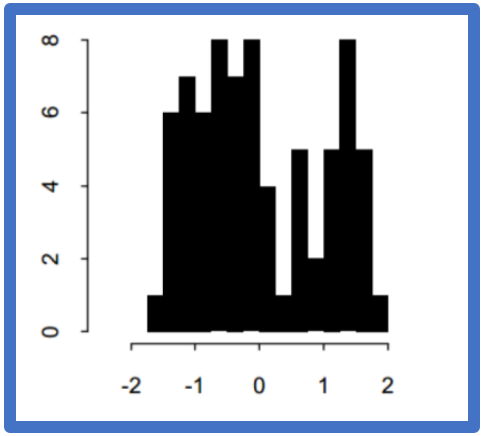


We study the concept of matching in distributions based on dissimilarity indices

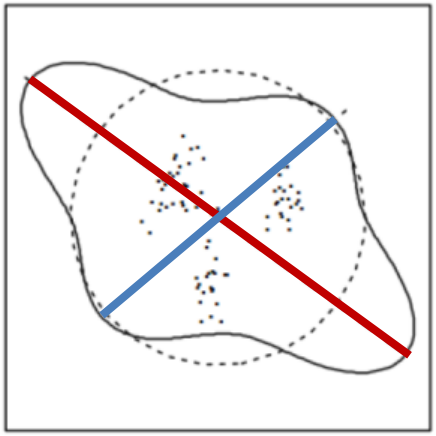
Indices to measure dissimilarity (among distributions)

Natural Hermite Index
(Cook, Buja, Cabrera, 1993)

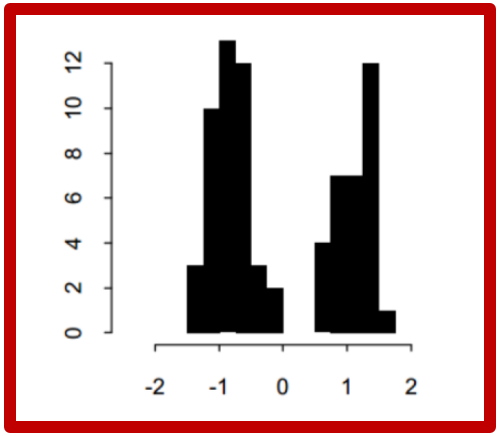
$$I^N = \int_{\mathbb{R}^d} \{f(\mathbf{y}) - \phi(\mathbf{y})\}^2 \phi(\mathbf{y}) d\mathbf{y}$$



1-dim projection with local max of NH index



First two dimensions of the flea beetle data



1-dim projection with global max of NH index

Differential Natural Hermite Index

- Analyzing experimental data, we often deal with differential experiments.
- Scientists may not care that the experiment “works” but that it “works better than control” or another treatment.
- The objective of Differential PP is is to find projections that maximize the difference between two or more distributions or treatments.

Differential Natural Hermite index for *k* populations

Definition: Differential Natural Hermite dissimilarity for *k* *d*-dimensional distributions $f_1(x), \ldots, f_k(x)$.

Let $f_1(x), \ldots, f_k(x)$ be a set of *k* density functions and let

$f(x) = \frac{w_1 f_1(x) + \cdots + w_k f_k(x)}{w_1 + \cdots + w_k}$ be the average. In many cases $w_1 = \cdots = w_k = 1$

For every pair of densities $f_i(x)$, $f_j(x)$ the differential Natural Hermite dissimilarity

with respect to $f(y)$ is defined by : $d_f(f_i, f_j) = \left| \int_{\mathbb{R}^d} [f_i(x) - f_j(x)]^2 f(x) dx \right|^{\frac{1}{2}}$

Differential Natural Hermite index for *k* populations

Proposition 1. The Differential Natural Hermite dissimilarity has the properties of a distance.

(i) $d_f(f_i, f_j) \geq 0$

(ii) $d_f(f_i, f_j) = 0$ is zero when $f_i(x) = f_j(x)$ for all x, except in a set of probability zero under f.

Suppose that S be a set where $f_i(x) \neq f_j(x)$ and $\int_S f(x)dx > 0$.

Then, for all $x \in S$ $[f_i(x)-f_j(x)]^2 > 0$ and $d_f(f_i, f_j) > 0$

(iii) $d_f(f_i, f_j) = d_f(f_j, f_i)$

(iv) $d_f(f_i, f_j) \leq d_f(f_j, f_k) + d_f(f_k, f_j)$

For all x, by the triangle inequality $|f_i(x) - f_j(x)| \leq |f_j(x) - f_k(x)| + |f_k(x) - f_j(x)|$. Therefore, the proof of (iv) follows from by proof of the standard triangular inequality. Therefore d_f is a distance.

Comparing multiple populations

For comparison of $k>2$ populations, we define the criterion

$$C = \sum_{i<j} d_f^2(f_i, f_j)$$

this would require the evaluation of $k(k\text{-}1)/2$ integrals.

However, in Weigle, Cabrera (2023) it was shown that:

For simplicity assume $w_1 = \cdots = w_k = 1$

Preposition: Given $f_1(y), \ldots, f_k(y)$ and $f(y) = \frac{f_1(y) + \cdots + f_k(y)}{k}$ then,

$$\sum_{i<j} d_f^2(f_i, f_j) = k \sum_i d_f^2(f_i, f)$$

The proof is very similar to show that suppose we observe $x_1, \ldots, x_k$, and calculate $\bar{x}$.

Then $\sum_{i \neq j}(x_i - x_j)^2 = 2k \sum_i (x_i - \bar{x})^2$ is sometimes used to compute the sample variance.

Propensity Index and Other indices

- Another simple way of defining an index that would be used for the Controls Augmentation process is to to use some function of the propensity scores.
- Suppose the dataset has a variable called Treatment: (Control=0, Treated=1) and a vector of covariates *X*.

$$h_{PS}(X) = P(Treatment = 1|X)$$

- We introduce a Propensity Index as a function $f(h_{PS}(X))$ for example

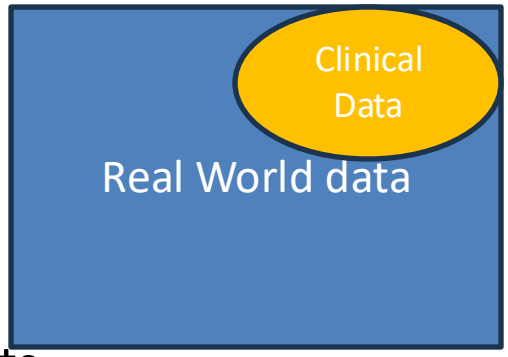
$$I_{PS}(X) = Var(h_{PS}(X))$$

- When $I_{PS}(X)$ is zero it means the propensity score function is constant and hence the population two populations are identical.
- The function $h_{PS}(X)$ is often estimated using a Super-Learner or in simple cases logistic regression etc.

Other similar indices could be derived from LDA, SVM, deep learning,…

Genetic algorithm to match populations
Preprocessing the real-world data

We want to augment a clinical data with real-world controls. Instead of matched pairs we will match the distributions by minimizing one index: PS index or NH index.



The starting real-world dataset is trimmed to a subset covering only the clinical data. This is an algorithm for this:

(i) Apply Fisher-Yates transformation to the variables.

(ii) Do a PCA and extract the first p ($p \leq 7$) principal components.

(iii) Apply again Fisher-Yates to the principal components.

Steps (i)-(iii) can be expressed as a transformation T: $\mathbb{R}^p \rightarrow \mathbb{R}^d$, where $z=T(x)$ is approximately normally distributed.

(i) Next the transformation $T(x)$ obtained from steps 1-3 is applied to the real-world data $z^* = T(x)$.

(ii) Compute the convex hull C_H of the transformed clinical study dataset and use it to discard the transformed real-world data that follows outside of C_H

Genetic algorithm to match populations
Preprocessing the real-world data

- Fisher-Yates: $y_i = FY(x_i) = \Phi^{-1}(\frac{Rank(x_i)}{n-1})$, where $\Phi(\mathrm{x})$. is the standard normal cdf.
- Suppose *X* a binary variable with 1% of 1's 99%0's *Y*=*X*/sd(X)

X=0	*X*=1	sd(*X*)	*Y*=*X*/sd(*X*)	*Z*=FY(*X*)	SD(*Z*)
99%	1%	0.0995	(0, 10.05)	(-0.031,2.23)	0.353
98%	2%	0.14	(0, 7.14)	(-0.038,2.16)	0.375
97%	3%	0.17	(0, 5.86)	(-0.044,2.09)	0.394
96%	4%	0.19	(0, 5.10)	(-0.050,2.04)	0.410

- *X* appear to minimize the contribution of binary variables.
- If transform *X* into *Y* some binary variables will create outliers or leverage points in the modeling.
- *Z* seems acceptable as it does not create leverage points and does not minimize the contribution of binary variables.

Preprocessing the real-world data

The starting real-world dataset is trimmed to a subset covering only the clinical data. This is an algorithm for this:

(i) Apply Fisher-Yates transformation to the variables.

(ii) Do a PCA and extract the first p ($p \leq 7$) principal components.

(iii) Apply again Fisher-Yates to the principal components.

Steps (i)-(iii) can be expressed as a transformation T: $\mathbb{R}^p \rightarrow \mathbb{R}^d$, where $z=T(x)$ is approximately normally distributed.

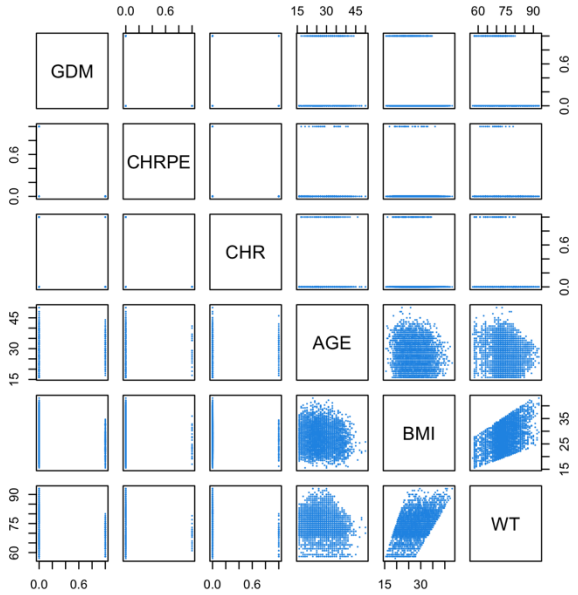
(i) Next the transformation $T(x)$ obtained from steps 1-3 is applied to the real-world data $z^*= T(x)$.

(ii) Compute the convex hull C_H of the transformed clinical study dataset and use it to discard the transformed real-world data that follows outside of C_H

An abruption study

A study on preventing placenta abruption on women in NJ (PACER,2023) . Want to use real world controls from pregnancy database of New Jersey hospital births. We have 18 variables (25 features) in common between the clinical study and the real-world database. (Limitation)

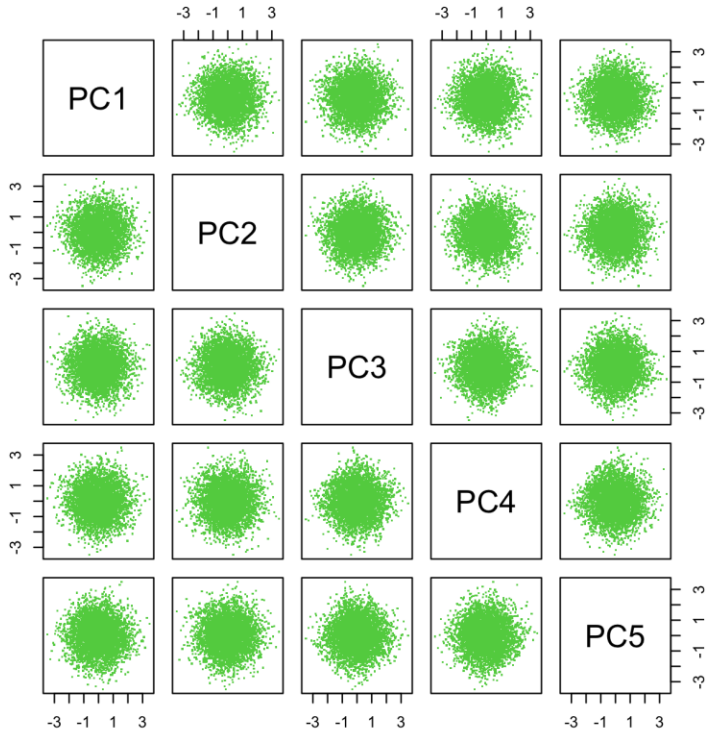
Scatter matrix of 6 variables of the 18 in data set.



Variables

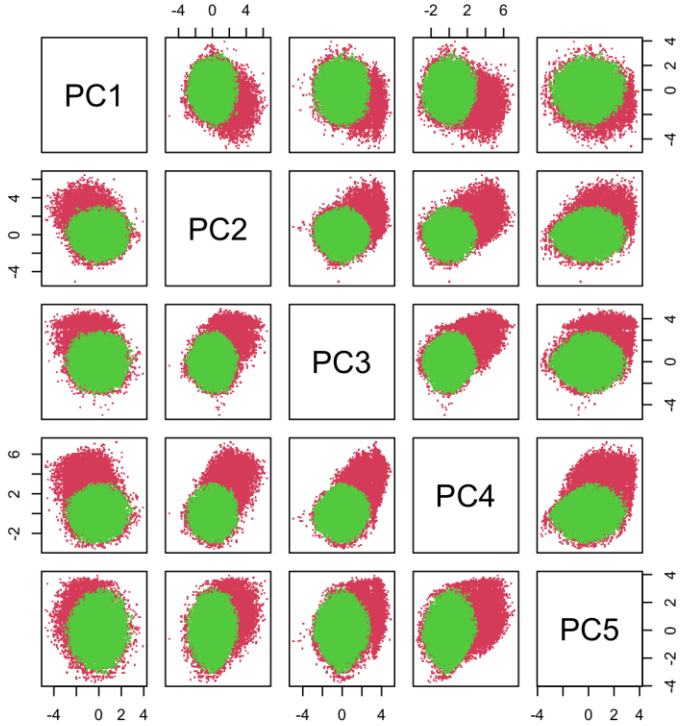
MONTH
PE_MILD
PE_SEVERE
REGION
RACE
PRE_DM
OLIGO
MARITAL
MULTIPLE
HOSPBEDR
HOSPOWN
GES_HYP
GDM
CHRPE
CHR
AGE
BMI
WT

Preprocessing a real-world (RW) dataset



5 principal components after Fisher-Yates transformation

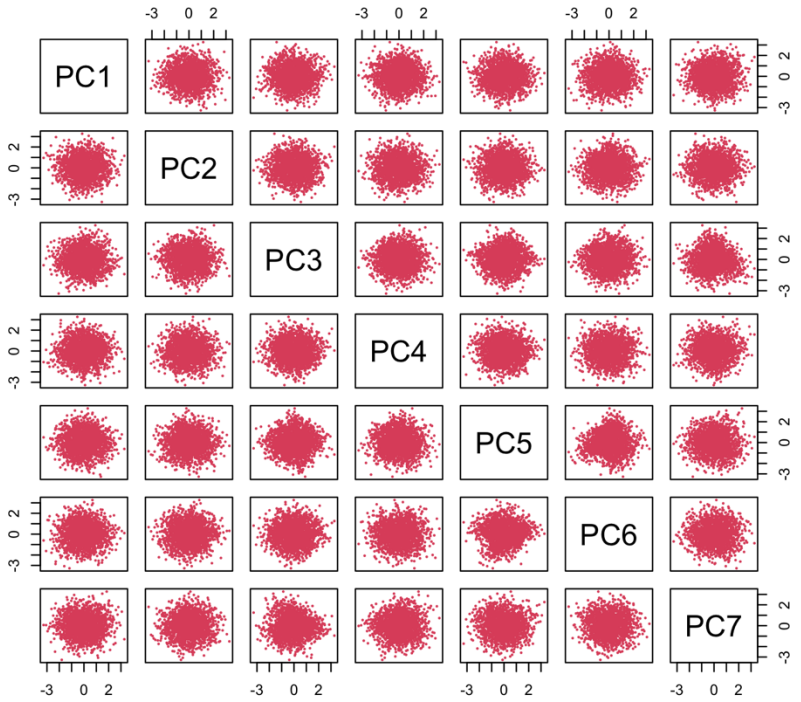
103K => 43K => 23K

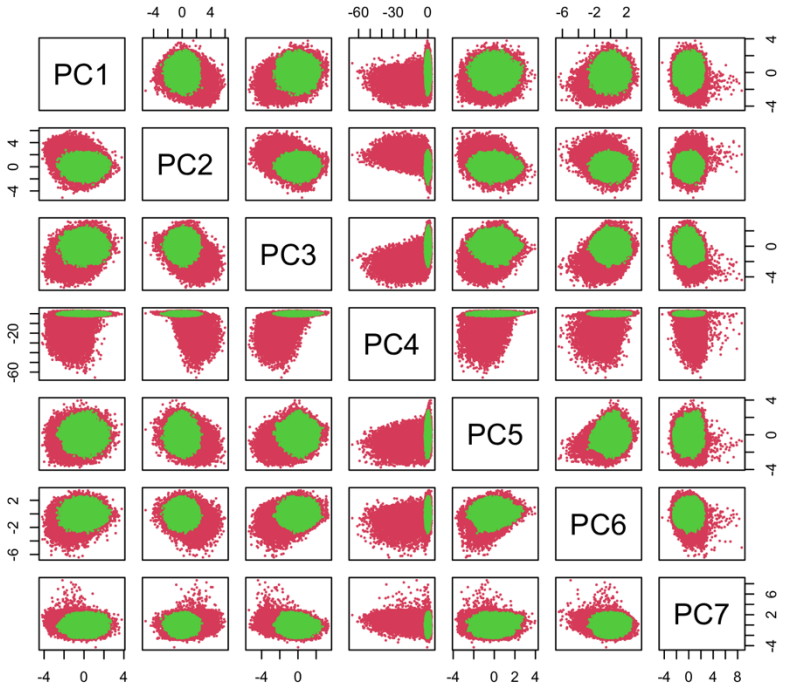


Same transformation applied to Real-World dataset.

In red is the subset or the RW outside the clinical study domain

Preprocessing a real-world (RW) dataset





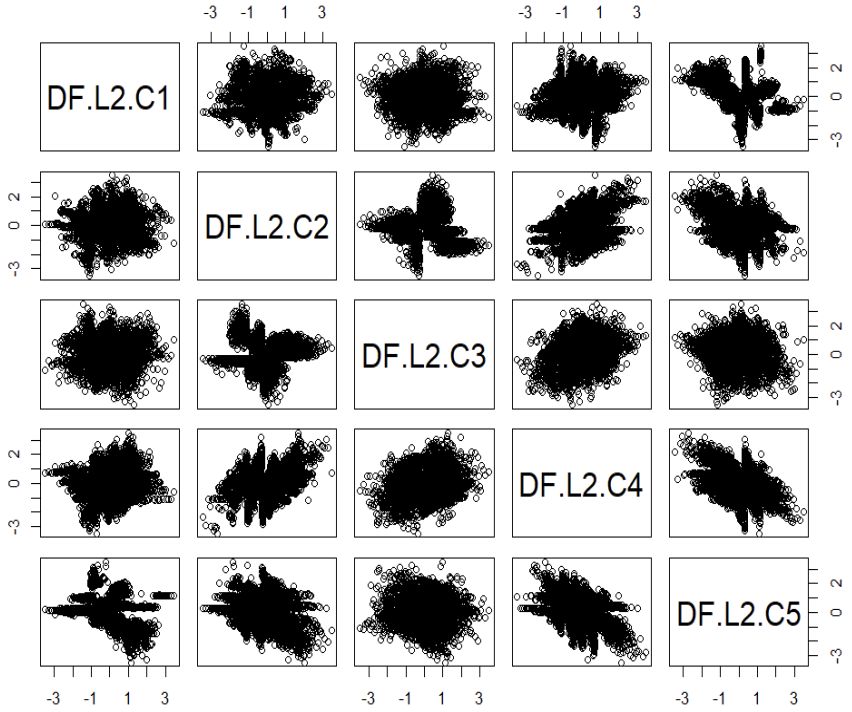
7 principal components after Fisher-Yates transformation

103K => 43K => 19K

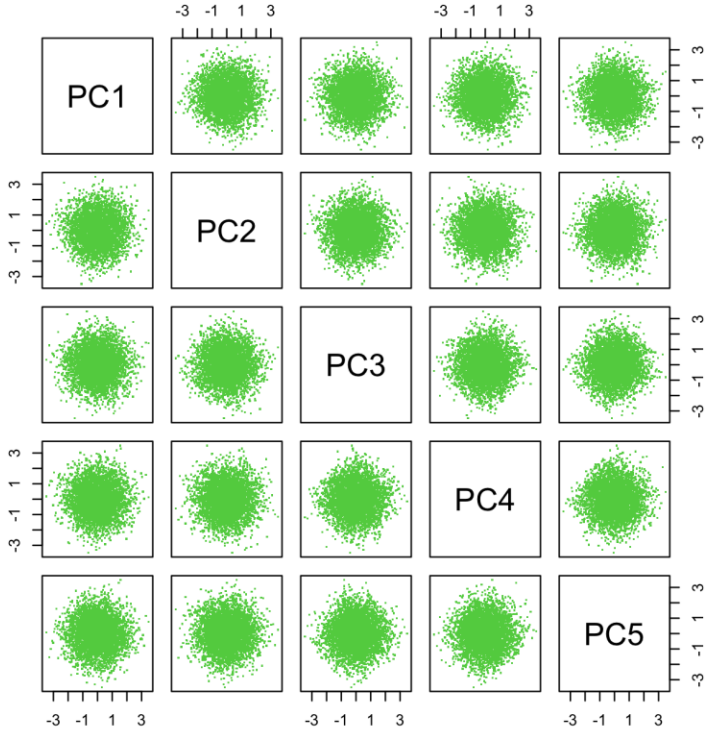
Same transformation applied to Real-World dataset.

In red is the subset or the RW outside the clinical study domain

Preprocessing a real-world (RW) dataset



Using Autoencoder



Using Fisher-Yates/ PCA

A Genetic algorithm for Augmenting control groups from real world data

Assume a trial comparing a test drug (A) vs a control (B), with n_1 and n_2 subjects assigned to A and B, respectively ($n_1 >> n_2$)

Suppose we have a RWD (*R*) set of size $M >> n_1$, and we wish to augment n_2 by drawing *m* observations from *R*.

Let *Y* denote the outcome variable of interest, and *X* a d-dimensional vector of covariates.

Goal: Find *m* observations from R that, combined with the n_2 controls in B, are 'similar' to the n_1 observations in A, the treatment group, according to some optimality criterion.

A Genetic algorithm for randomization

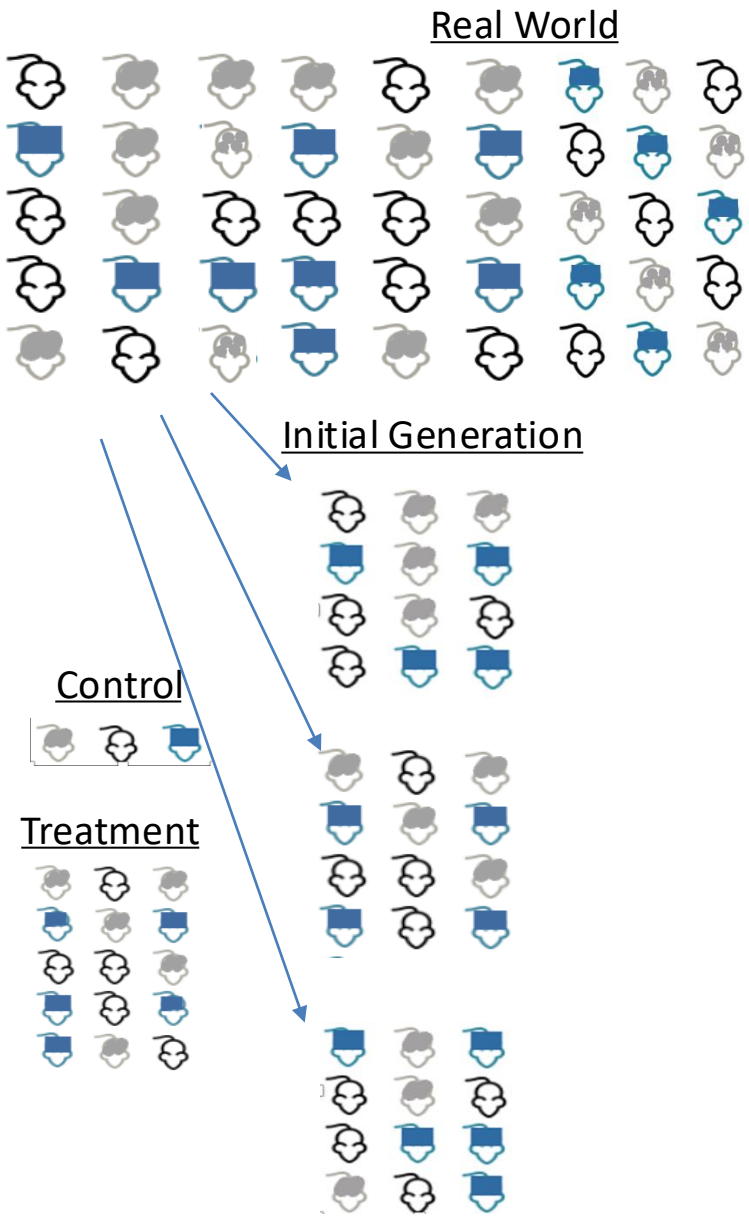
Real World

Genetic Alg. Fitness Criteria:
Propensity Index, Nat. Hermite Index, Ave CV^{-1},

Control

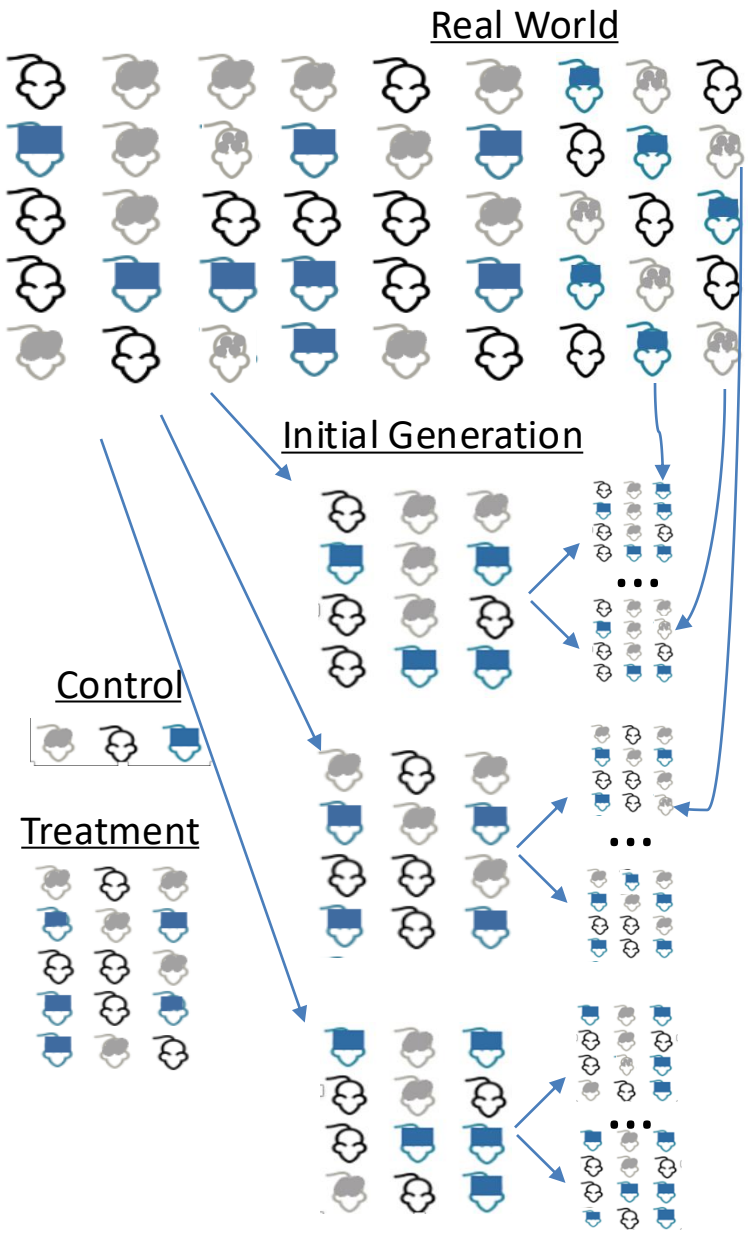
Treatment

A Genetic algorithm for randomization



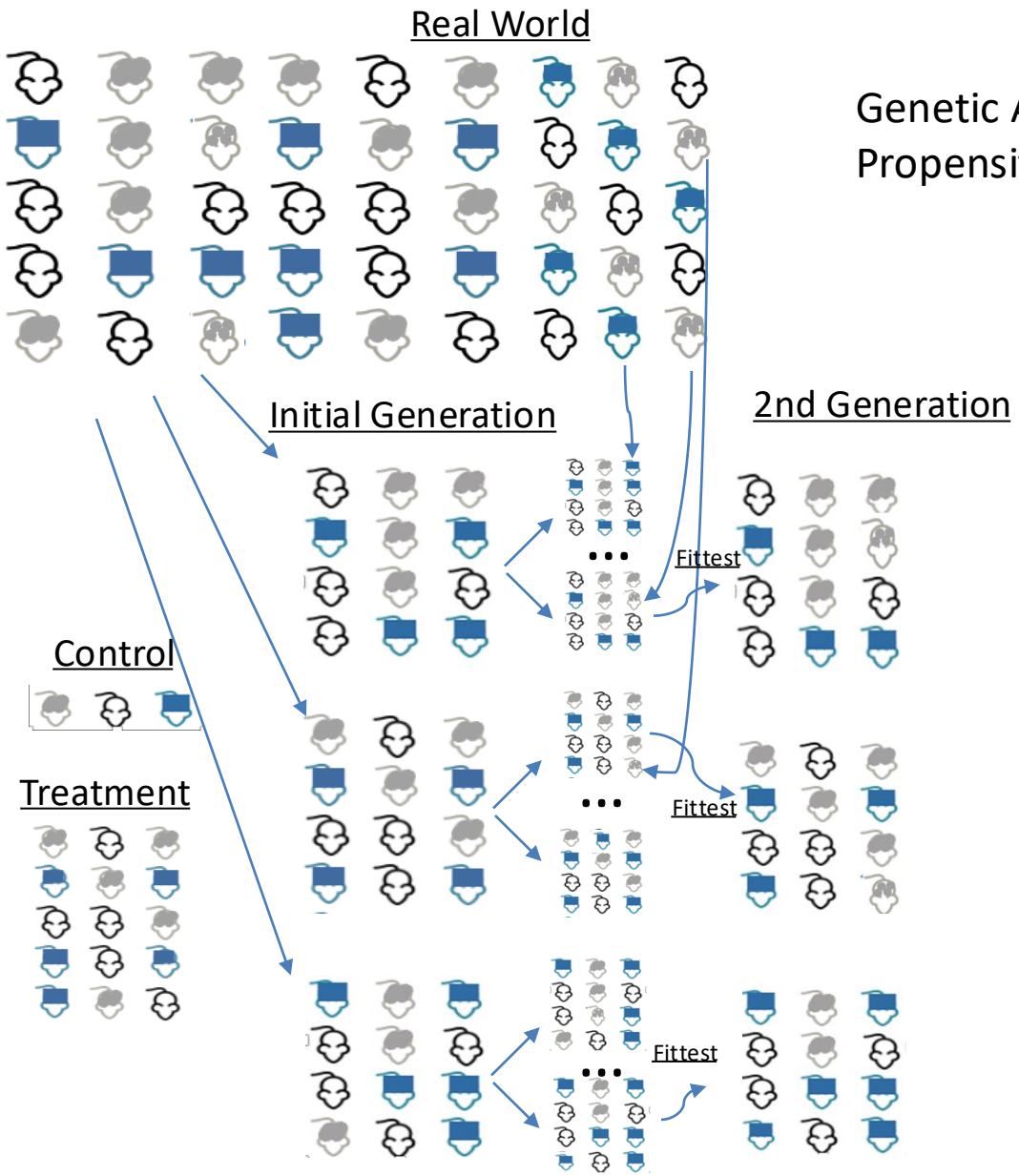
Genetic Alg. Fitness Criteria:
Propensity Index, Nat. Hermite Index, Ave CV^{-1},

A Genetic algorithm for randomization



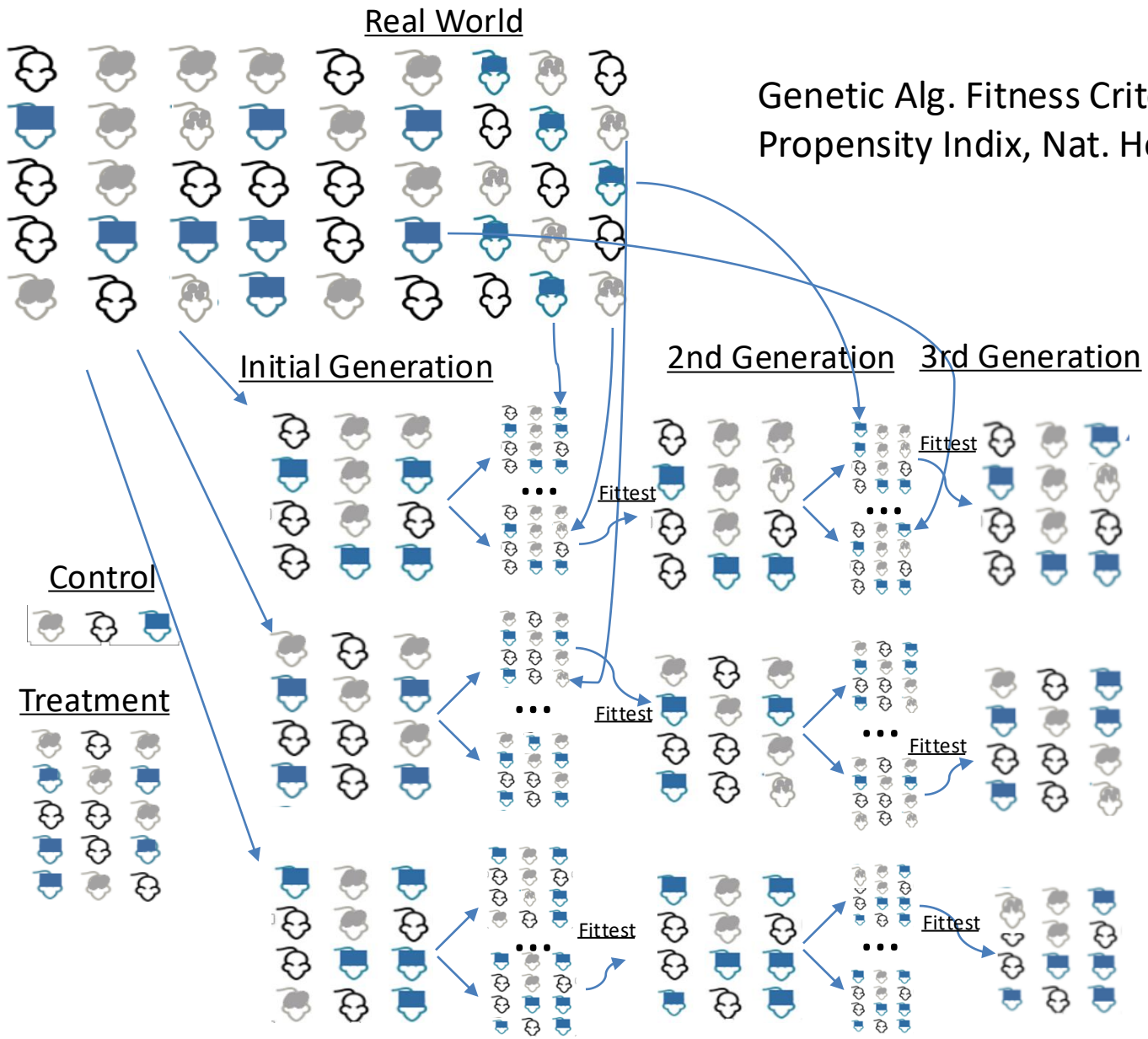
Genetic Alg. Fitness Criteria:
Propensity Index, Nat. Hermite Index, Ave CV^{-1},

A Genetic algorithm for randomization



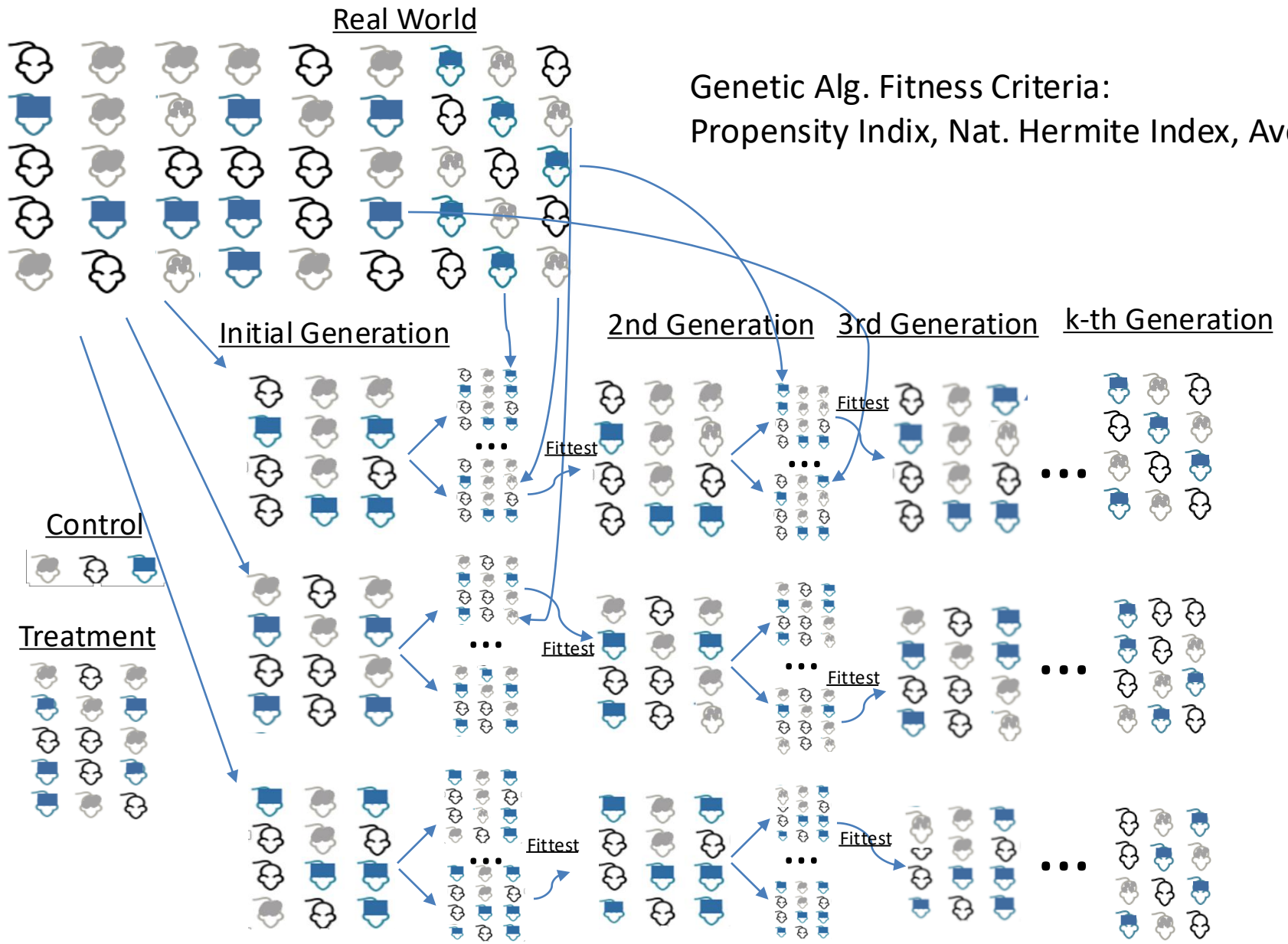
Genetic Alg. Fitness Criteria:
Propensity Index, Nat. Hermite Index, Ave CV^{-1},

A Genetic algorithm for randomization

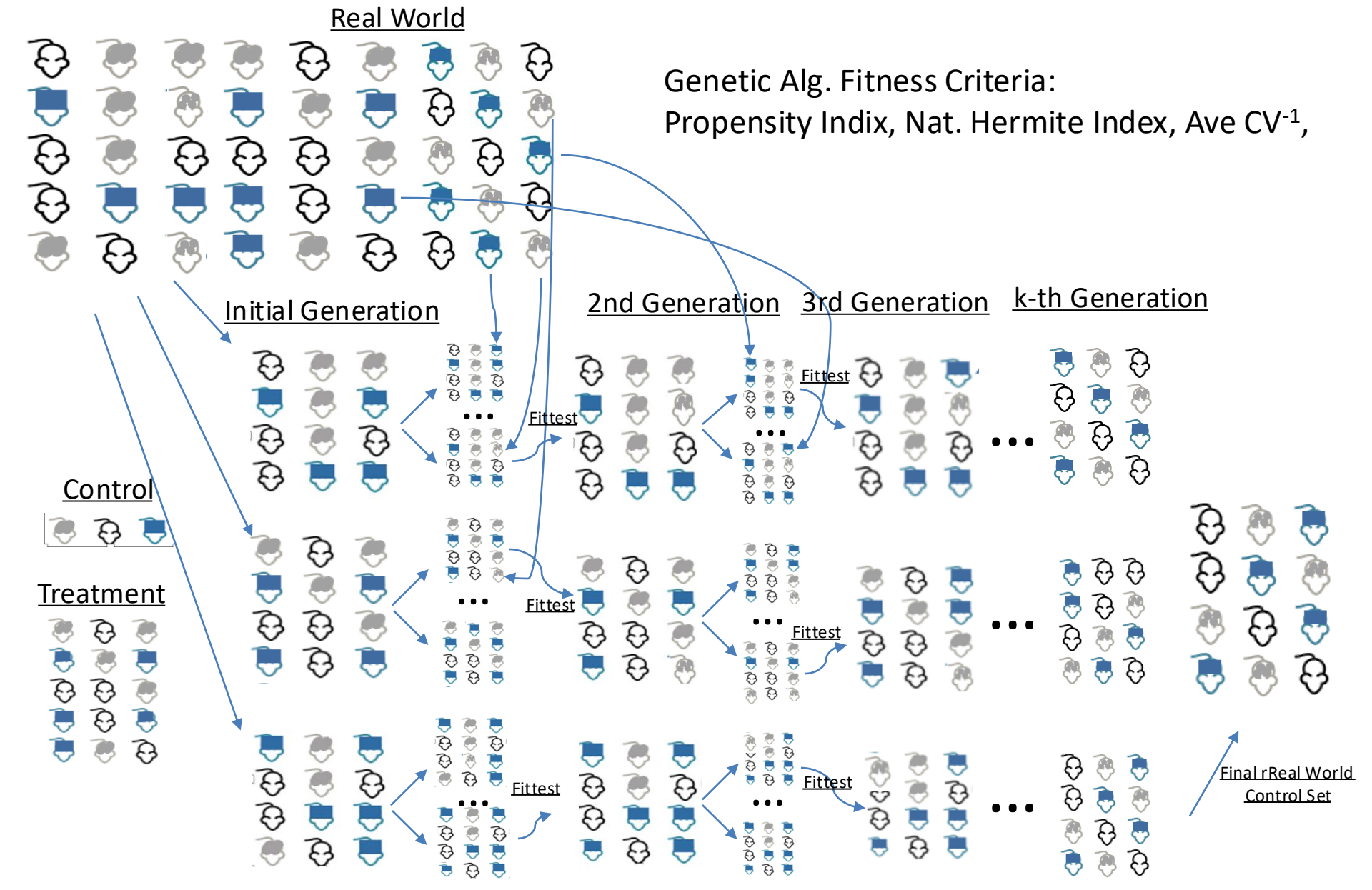


Genetic Alg. Fitness Criteria:
Propensity Index, Nat. Hermite Index, Ave CV^{-1},

A Genetic algorithm for randomization



Genetic algorithm to match populations



BACKGROUND

- In PP we optimize and index by observing index values in a neighborhood of the current projection and choosing the next projection
- In GA the Index optimized by mutations forming new generations followed by natural selection.
- We need to define what corresponds to a mutation in our context.
- The simplest way is to switch a pair between R and B.
- Super-Learner variability is an issue with stopping rule.
  - Simple stopping rule: k generation where the top configuration is the same and the index value was less than some.

A Genetic algorithm for Augmenting control groups from real world data

The proposed approach consists of the following steps:

1. Draw an initial k (e.g., k=10) sets of m observations from R the RWD. Compute corresponding k index measures $\{c_i\}$. Also calculate

$$C_1 = \min_{1 \le i \le k} c_i$$

For g=2, …, G

2. Interchange one observation from each initial set of m observations by randomly drawn an observation from the the RWD.
3. Compute the index measure for each augmented datasets.
4. Repeat the above s times (e.g., s=50), forming $k \times s$ replicates of n_2+m observations and $k \times s$ index measures (c_{gij} , i=1, ..k; j=1,…,s).

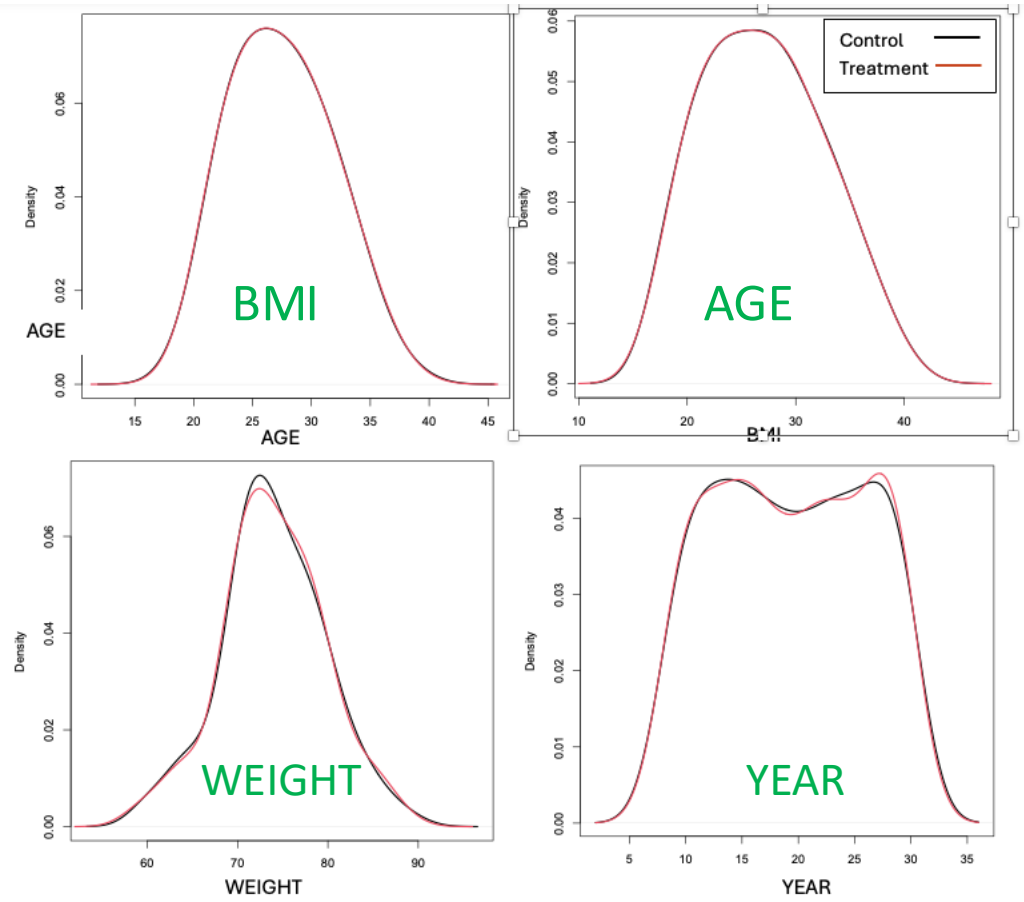
5. Select the k datasets with the smallest c_{gij} and compute C_g the smallest $\{c_{gij}, C_{g-1}\}$

6. Repeat steps 2-5 G times until $C_{g+1-\delta}$- C_g correspond to the same configuration and are the lowest (we used $\delta = 3$) and less than 0.05

Results of the Genetic algorithm applied to the Abruption datasets

After preprocessing and subsetting RW= 18992records, Clinical study: 1857 records.

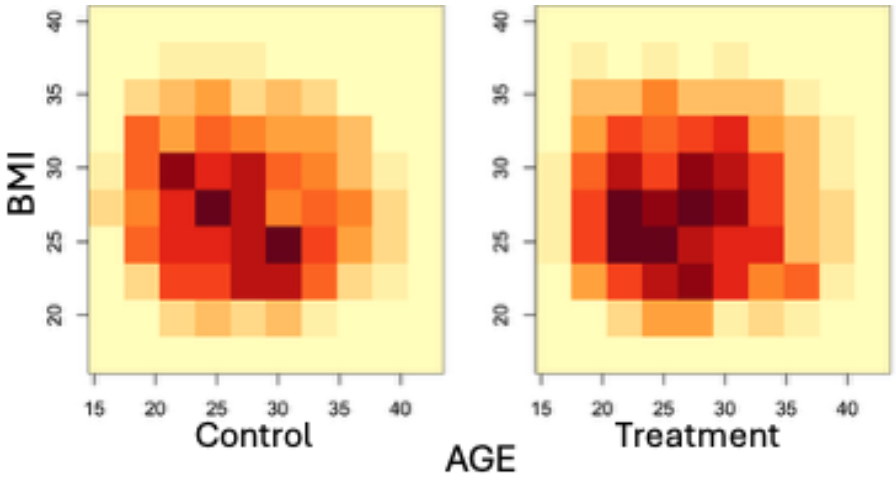
Results of Genetic algorithm For Continuous vars

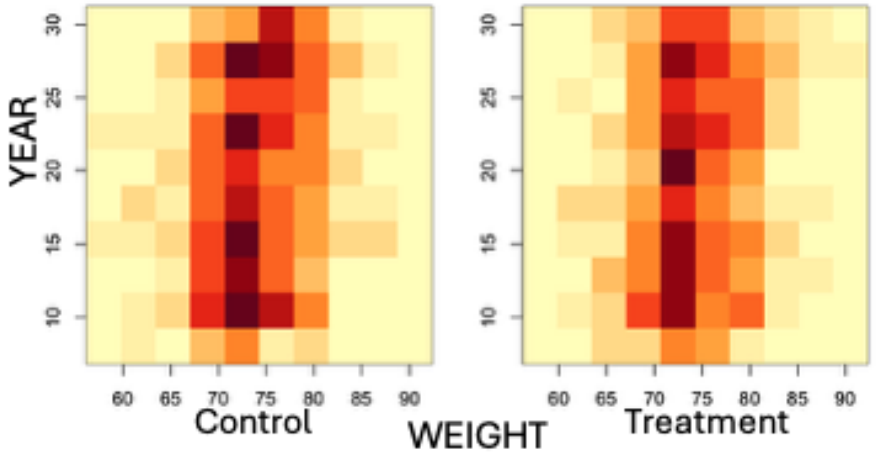


For Binary Vars the marginals were identical

Results of the Genetic algorithm applied to the Abruption datasets

Density estimators of the distributions of four continuous predictors in the treatment and control datasets. The controls dataset is a subset of the real-world dataset of controls.





Example: Simulation following the K. Hoffman TMLE tutorial model.

Ref: https://www.khstats.com/blog/tmle/tutorial

N Treatment group = 100, N Control group = 50,

Real World database: 50,000 controls.

No of Simulations = 150 datasets,

Stopping rule: if the same subset

after 15 generations the index was still quite big > 0.05

$W_1 \sim$ Bernoulli(p=0.2), $W_2 \sim$ Bernoulli(p=0.5), $W_3 \sim$ Round(Unif(2,7)), $W_4 \sim$ Round(Unif(0,4))

$$Treatment \sim PLogis(-0.2 + 0.2\,W_2 + \log(0.1W_3) + 0.3\,W_{Z4} + 0.2\,W_1W_4)$$

$$Y \sim Bern(prob = PLogis(-1 + A - 0.1W_1 + 0.2W_2 + 0.3W_3 - 0.1W_4 + \sin(0.1W_2W_4)))$$

Results

After running the GA for augmentation of controls for 150 datasets for 15 generations, 17 did not meet the convergence criteria.

Of the remanning 133 datasets that met the converge criteria and reached a small index value, a generalized linear model was fitted with the predictors and the p-value of treatment was recorded.

The same initial dataset was augmented with the initial random allocation of the GA and the same model was fitted.

GA augmentation	Random augmentation				Random augmentation			
	p>=0.01	p<0.01			p>=0.001	p<0.001		
p>=0.01	61	11	54%	p>=0.001	92	7	78%	
p<0.01	18	43	46%	p<0.001	12	22	26%	
	59%	41%			78%	22%		

Applications of GA, DNHI and PSI to randomization of animal studies

Animal studies:

- Divide the animal pool into groups that have similar distributions.
- Irini: Original algorithm minimizing inverse CV average.
- Same problem as optimal training and testing

Objective: Select a partition of the data to make the distribution of the groups as close as possible to eachother.

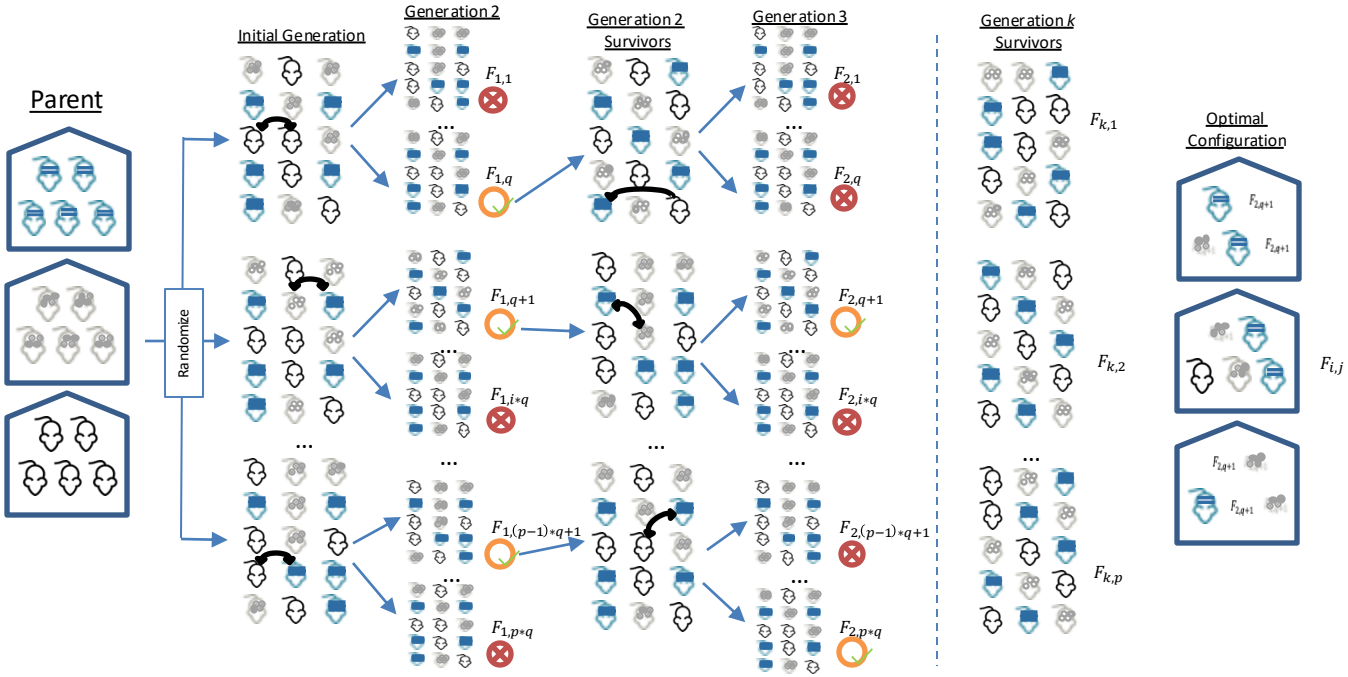
Applications of GA, DNHI and PSI to randomization of animal studies

Objective: *N* Animals to be randomized into *k* groups optimizing a criteria.

Algorithms: Exhaustive search, Genetic Alg.
Fitness function: Irini (Ave CV^{-1}), Hermite distance

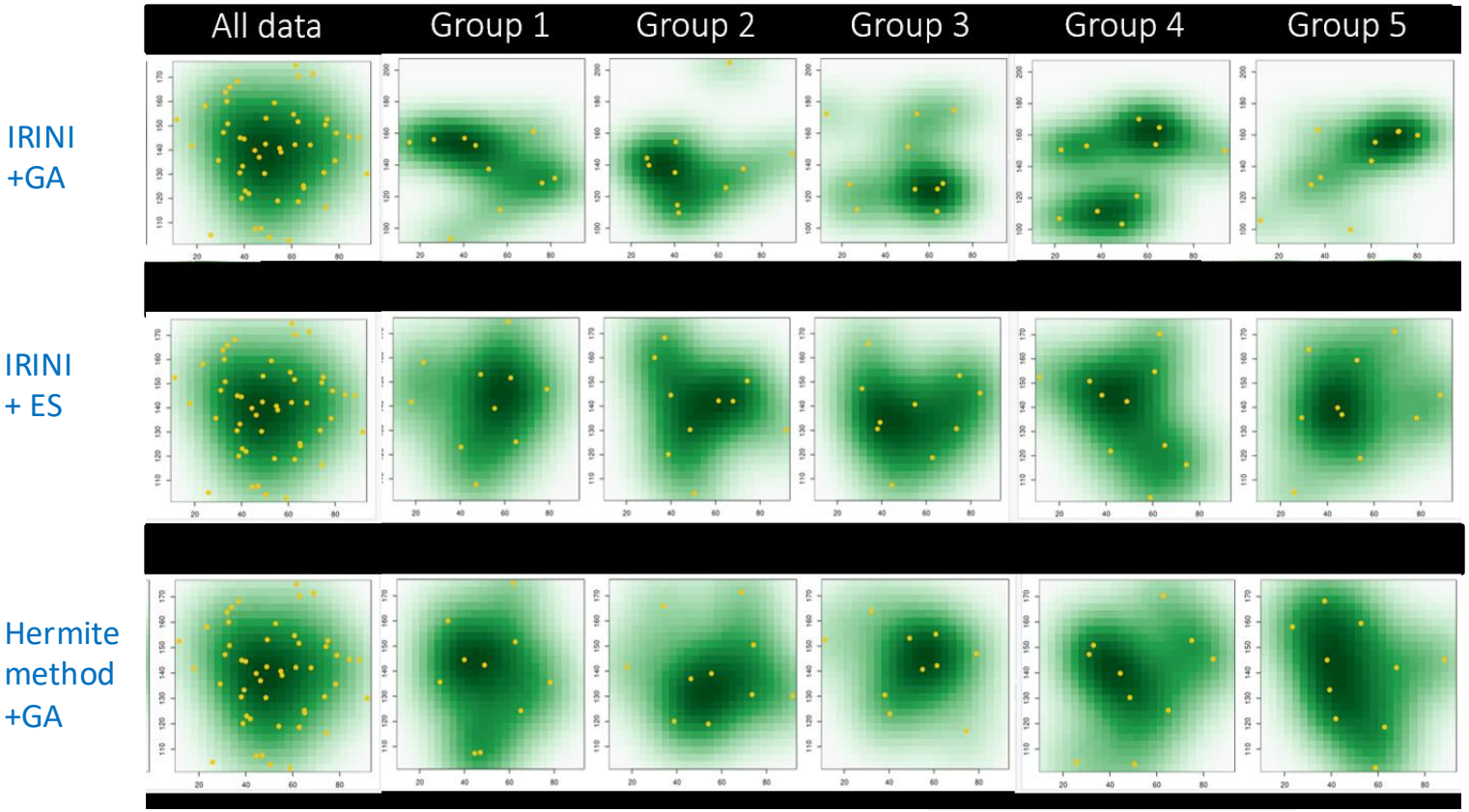
Genetic Algorithm:

Multiple generation + Survival of the fittest



Algorithm for sequential randomization to rebalance clinical studies

Example of results with Bivariate Data



Example: Simple Simulation

Dataset: 20 mice with two covariates Cov_1 and Cov_2, to be randomized into two groups of mice of size 10 each. Cov_1 and $Cov_2 \sim N(0,1)$

Method 1. Random allocation

Method 2. Genetic algorithm optimizing the Natural Hermite index

Model for generating *response:*

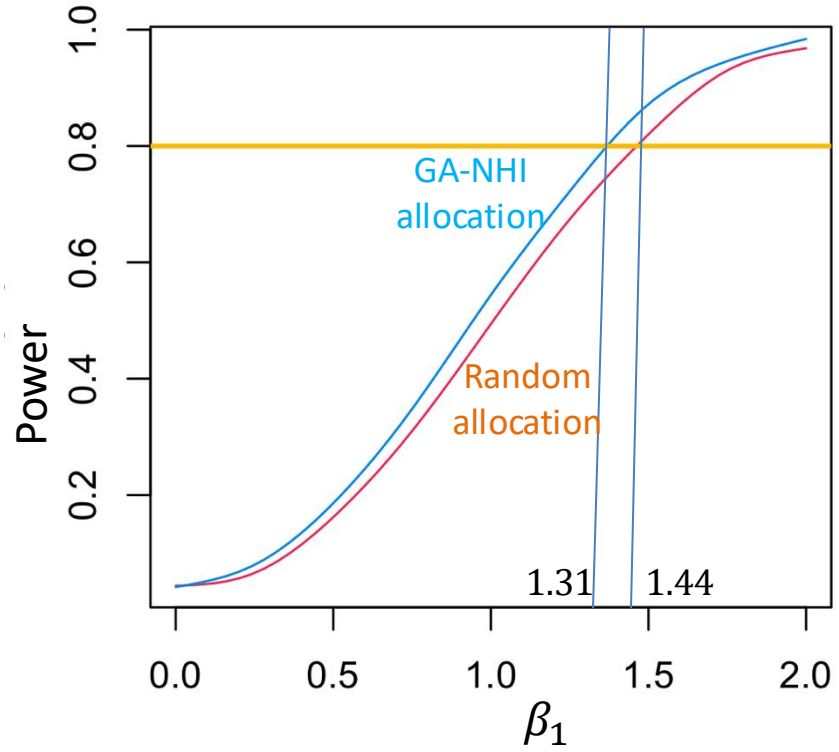
$$Y = \beta_1 Treatment + \beta_2 Cov_1 + \beta_3 Cov_2 + \varepsilon$$

Where

$\beta_1 = 0,0.25,0.5,0.75,1,1.25,1.5,1.75,2$

$\beta_2 = 1, \beta_3 = -1 \; and \; \sigma_\varepsilon = 1$

No of Simulations = 500 per value of β_1



Algorithm for sequential randomization to rebalance clinical studies

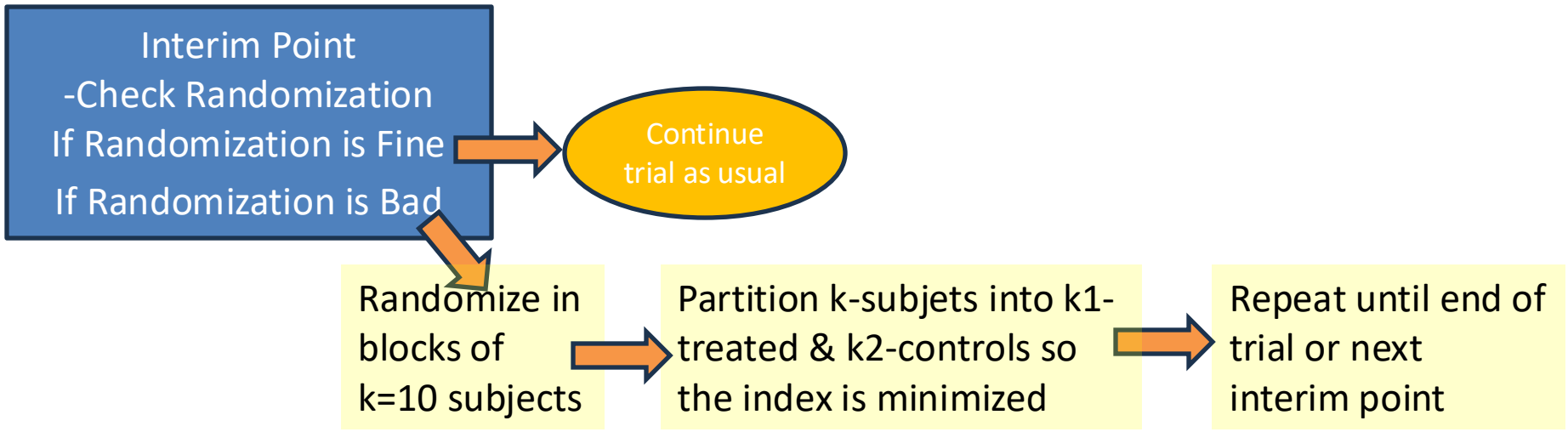
Clinical Trials get unbalanced (often loss of 25% of the subjects)

Loss usually not at random so Treatment and Control groups are unbalanced.

1. At Interim point check the samples balance.

- Check PS-index or NH index between treatment and control
- If the value is large compared to random split start Balancing pseudo-randomization

Balancing pseudo-randomization:



Conclusions and Further Work

- This work is still in its initial stages
- The results show modest gains, but I feel they are promising
- Need to improve many of the steps of the algorithm
- Need to find good real data examples that show the advantages of the method

References:

- Duan, Cabrera, Emir (2024). A New Projection Pursuit Index for Big Data. JCGS,
- Dastgiri, Duan, Cabrera (2025). Novel Machine Learning Approach to Differential Flow Cytometry Analysis base on differential Projection Pursuit. JBS
- Weigle, Cabrera, Sargsyan(2025) Randomization in pre-clinical studies: when evolution theory meets statistics. (under revision). JBS

Part III

Machine learning and deep learning methods for data augmentation and matching populations.

- Introduction.
- Issues on matching large populations.
- Data nuggets for compressing big data.
- Matching in distribution using data nuggets and applications to large clinical and epidemiological studies.
- Variational auto-encoder (VAE) for compressing data into a lower-dimensional "latent space".
- Applications of VAE to matching and control augmentation.

Issues on matching large populations

When one or both populations are very large the computational cost of the matching is very large.

To match distances we may have to compute N x M distances

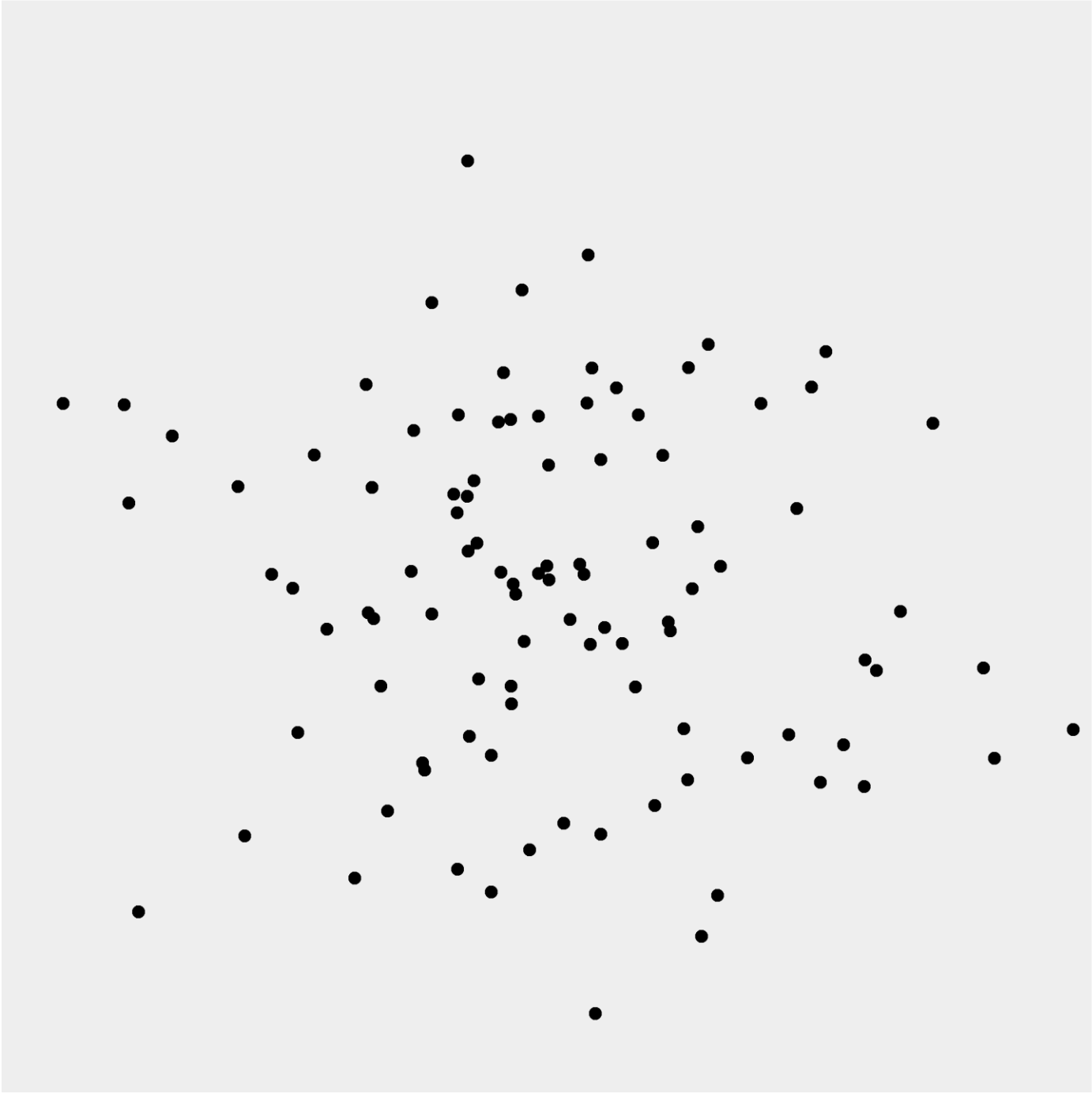
If $N = 10^6$ and $M = 10^4$ then $N \times M = 10^{10}$ distances.

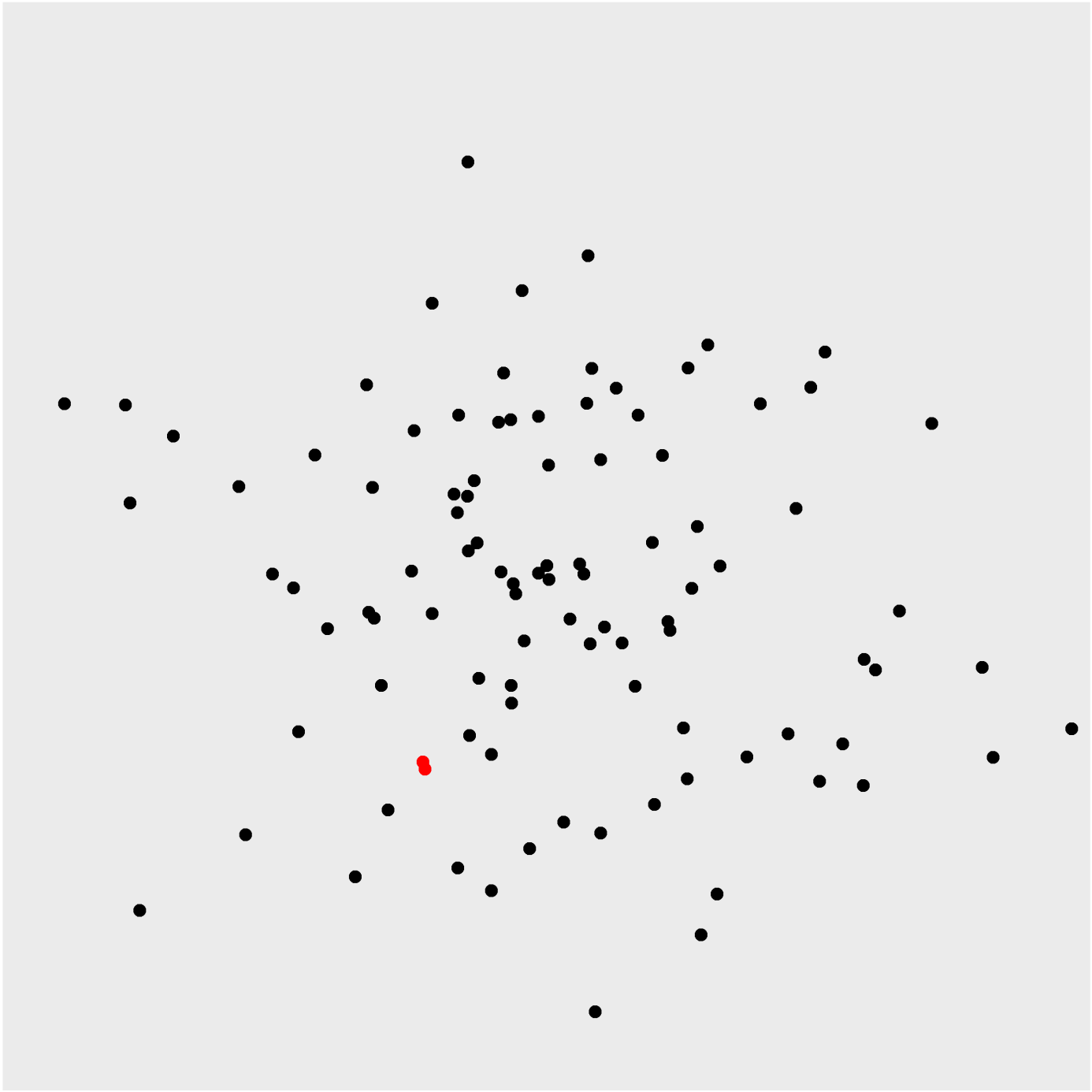
One idea is to use the datanugget method to compress the data into approximately 5000 nuggets each with a proportion of treated and controls.

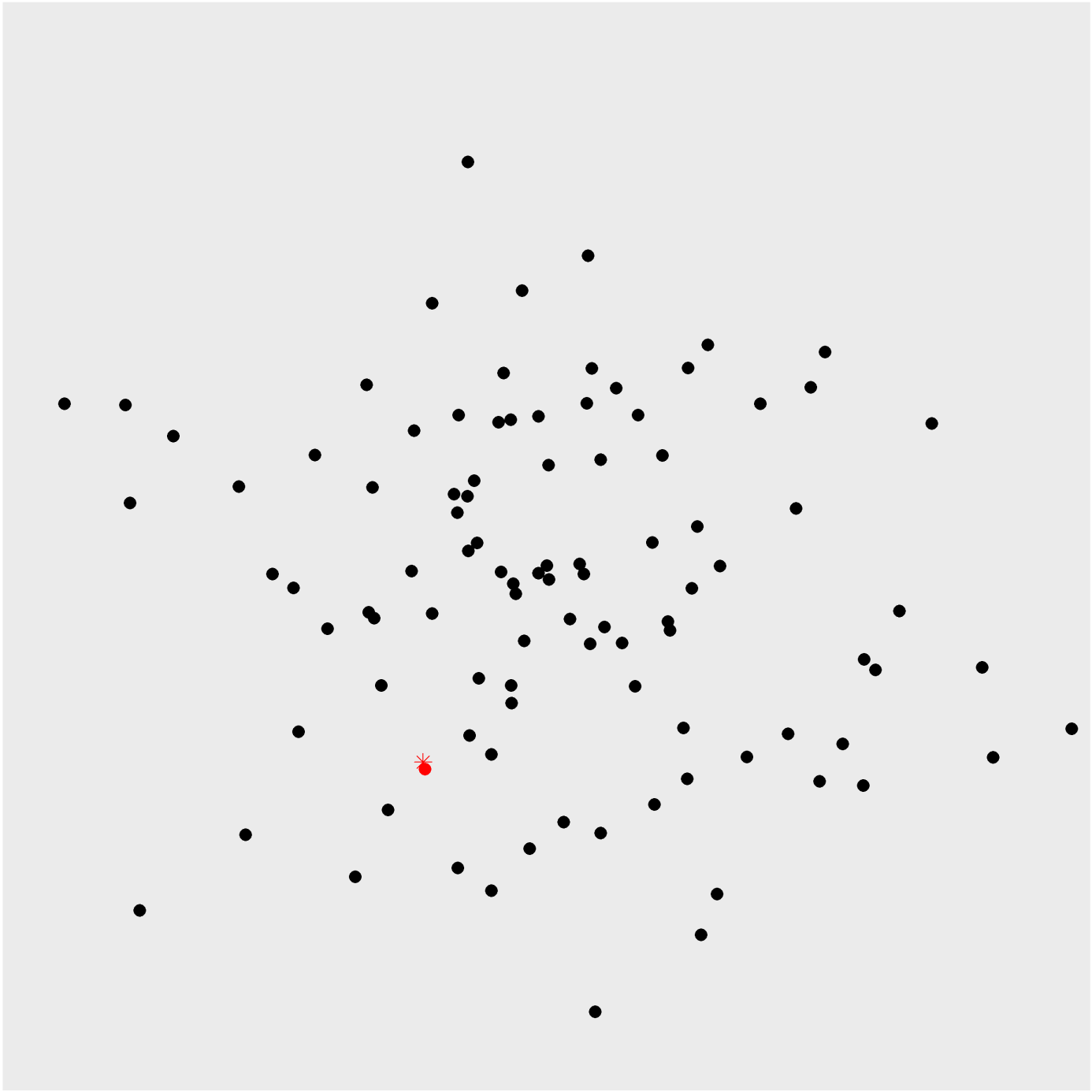
Data nuggets for compressing big data

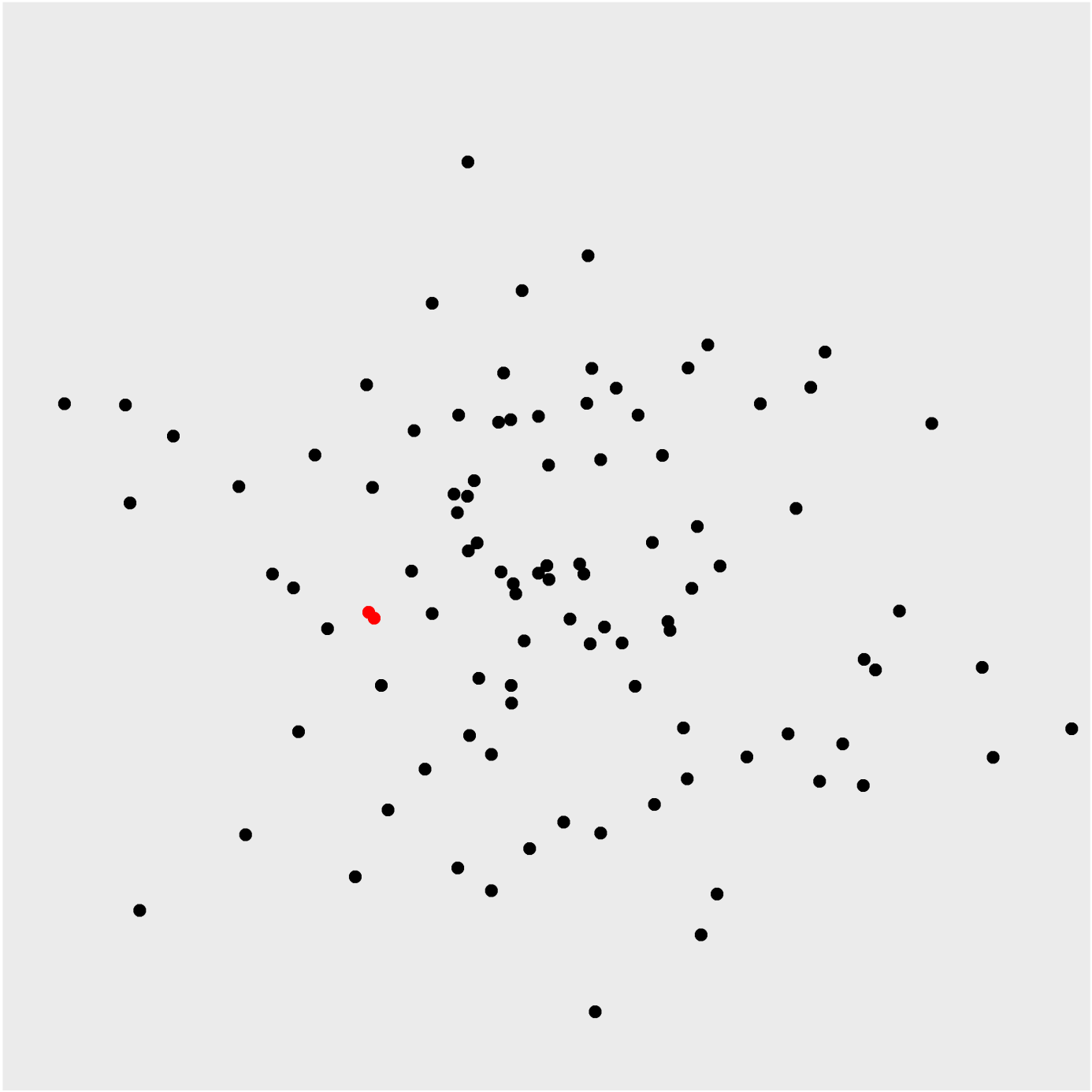
Data Nuggets: Compression algorithm for big datasets

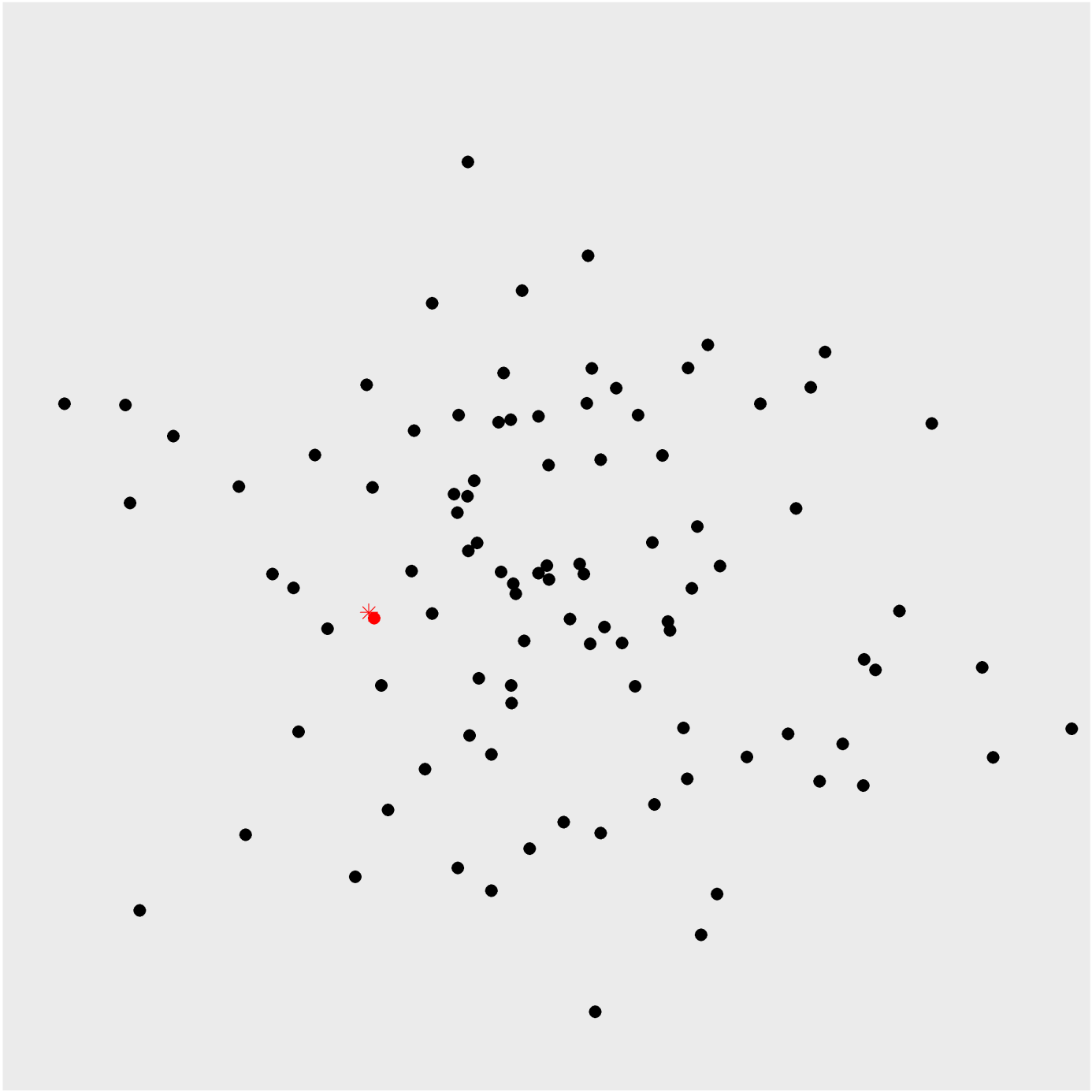
- Data nuggets method (published with my student Beavers et al. Cabrera, JCGS 2024) reduces a large dataset into a small collection of nuggets of data while maintaining the data structure, each containing center, weight, and scale parameters.
- R-packages in CRAN:
  - datanugget with more than 65000 downloads
  - Wcluster. 50000 downloads
  - PPbigdata 2000
- in github: DNAMR

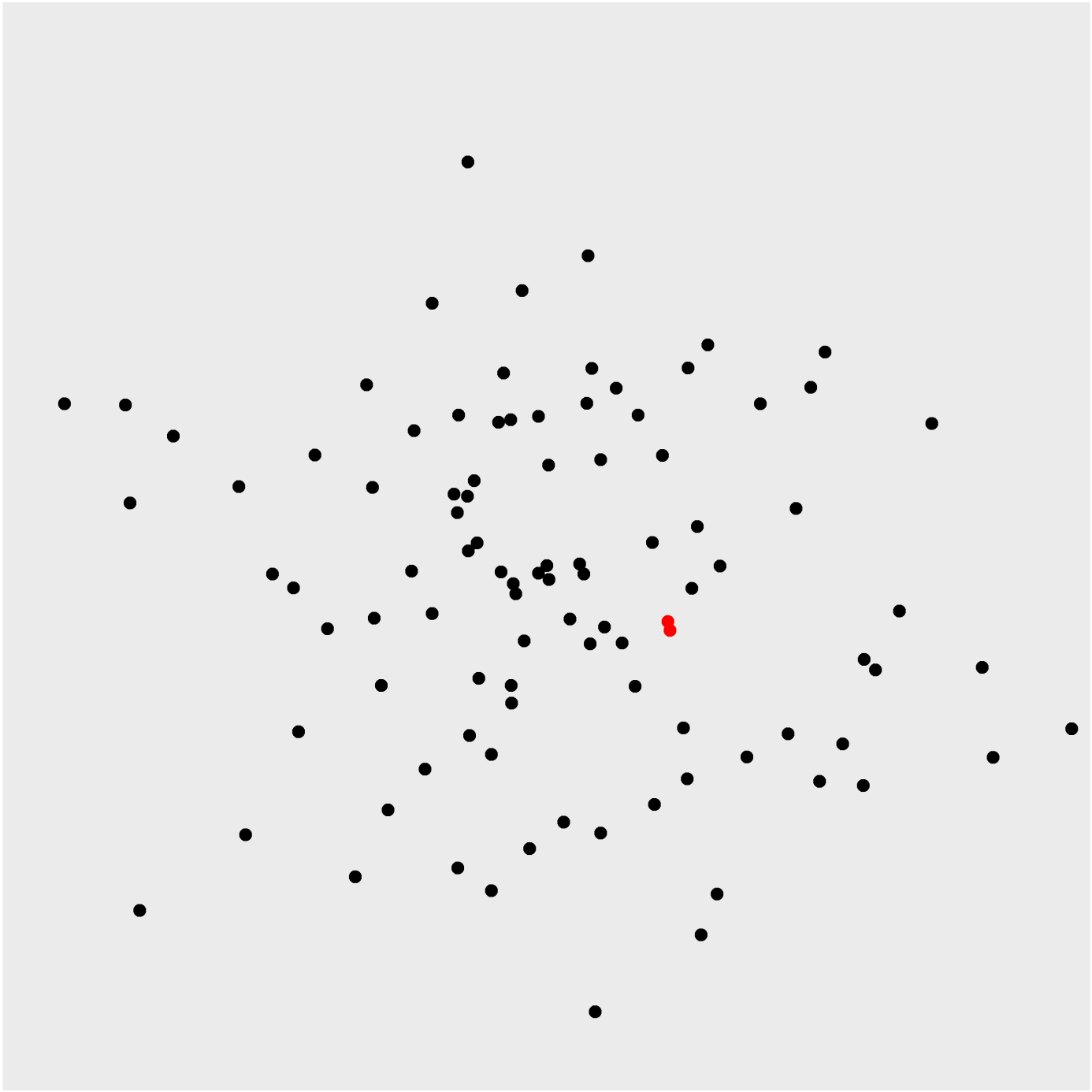


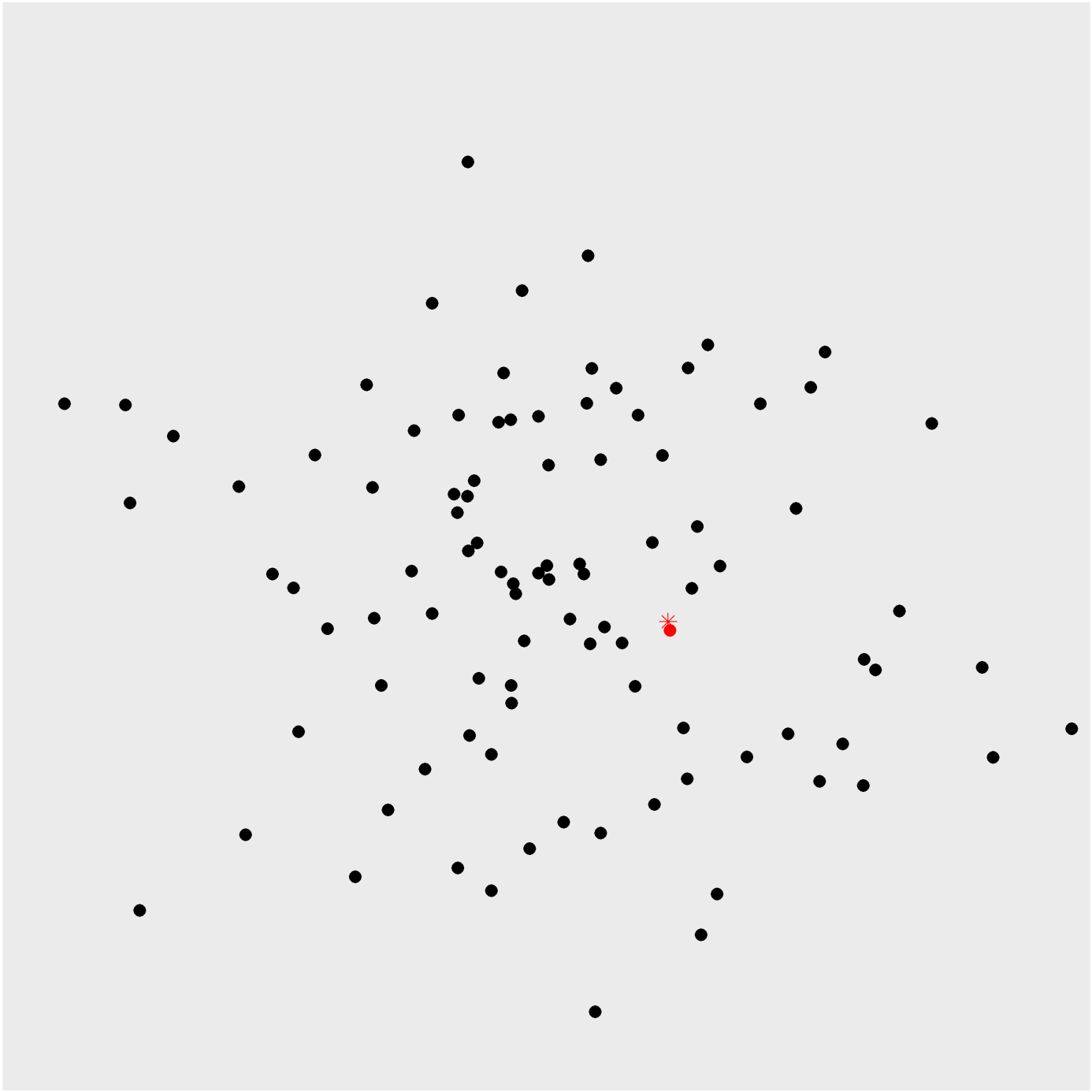


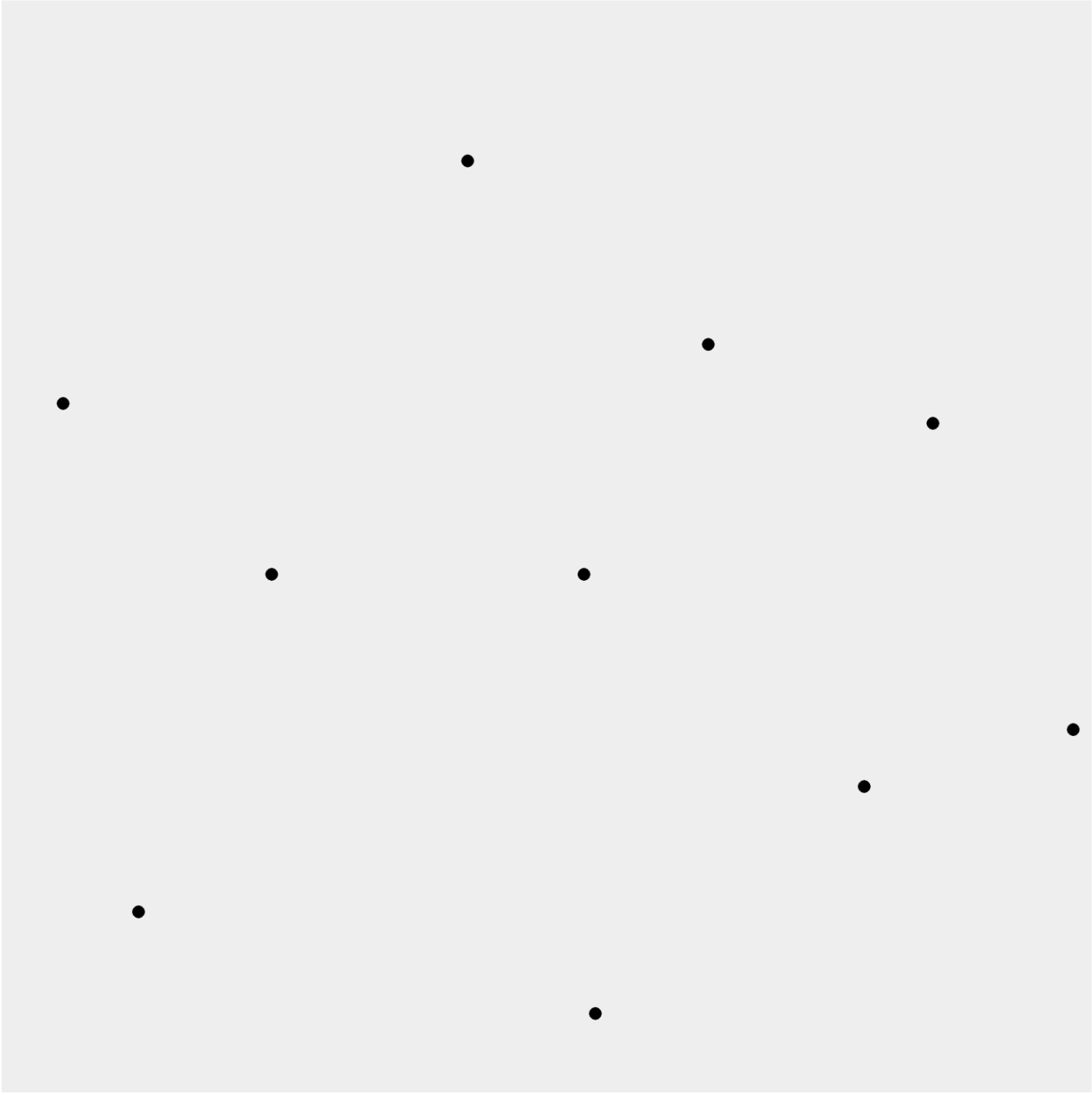


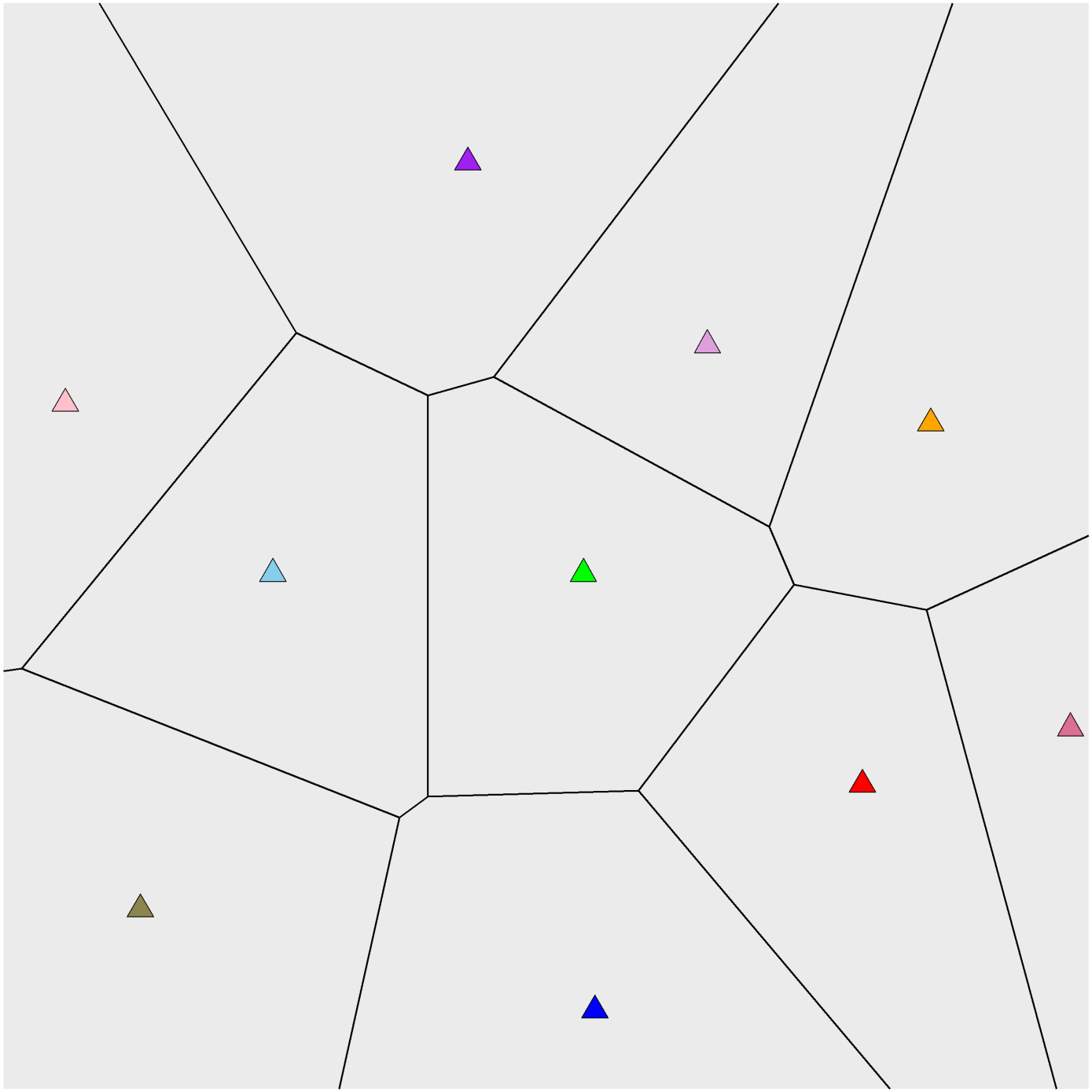


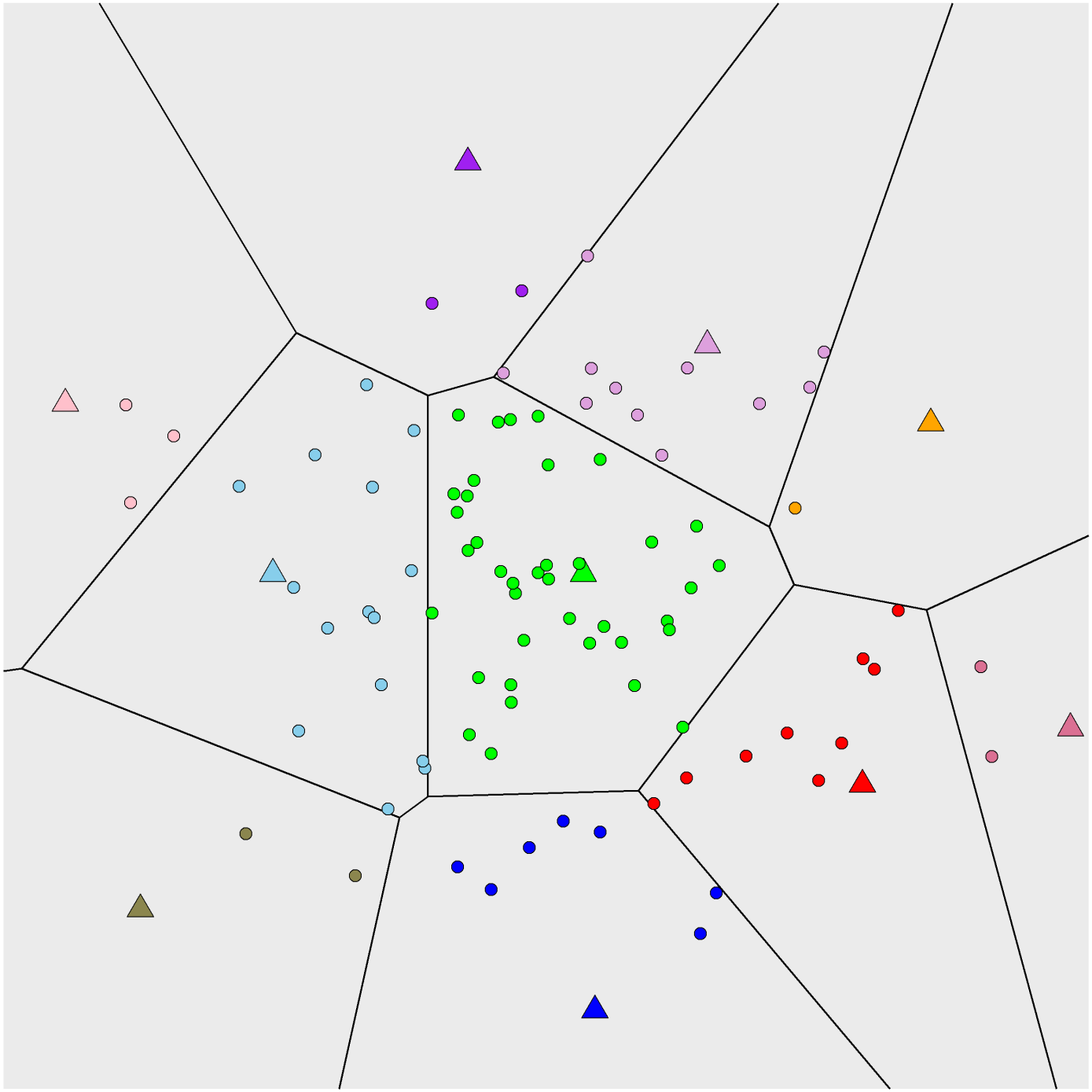










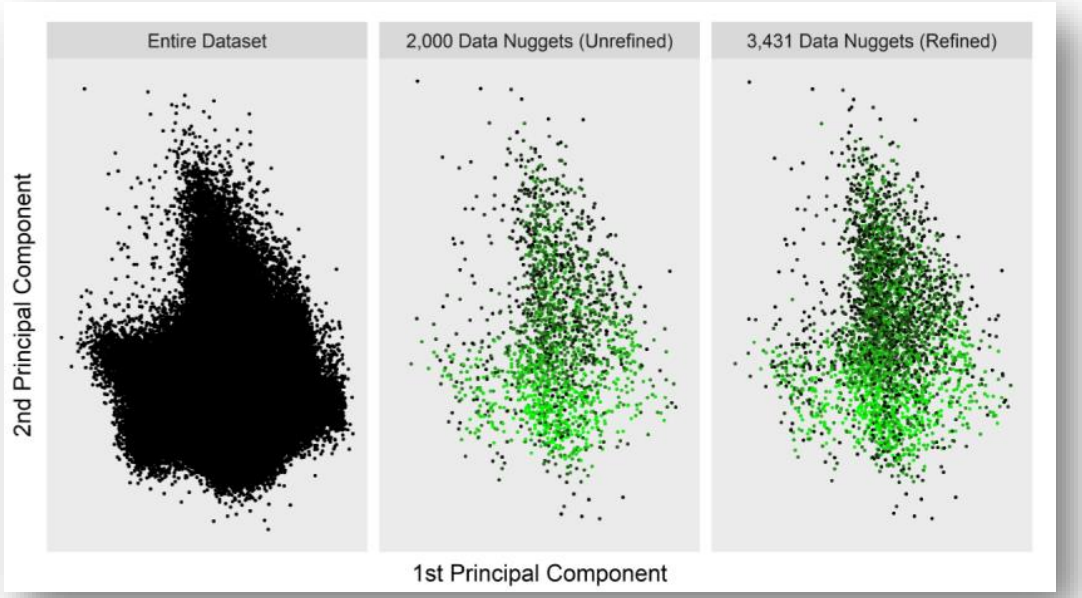


Introduction to Data Nuggets

For large N*P in the millions, we normally construct 2000-5000 data nuggets.

Each data nugget consists of :

Center: mean of observations inside the nugget
Weight: number of observations
Scale: $s_i = \text{Max}\{\text{diag}(\text{Cov}(\mathbf{X}_i))\}$ where $\mathbf{X}_j$ is the observations belonging to data nugget i. When weight is 1, $s_i = 0$.



Matching distributions using data nuggets and its Uses for large clinical and epidemiological studies

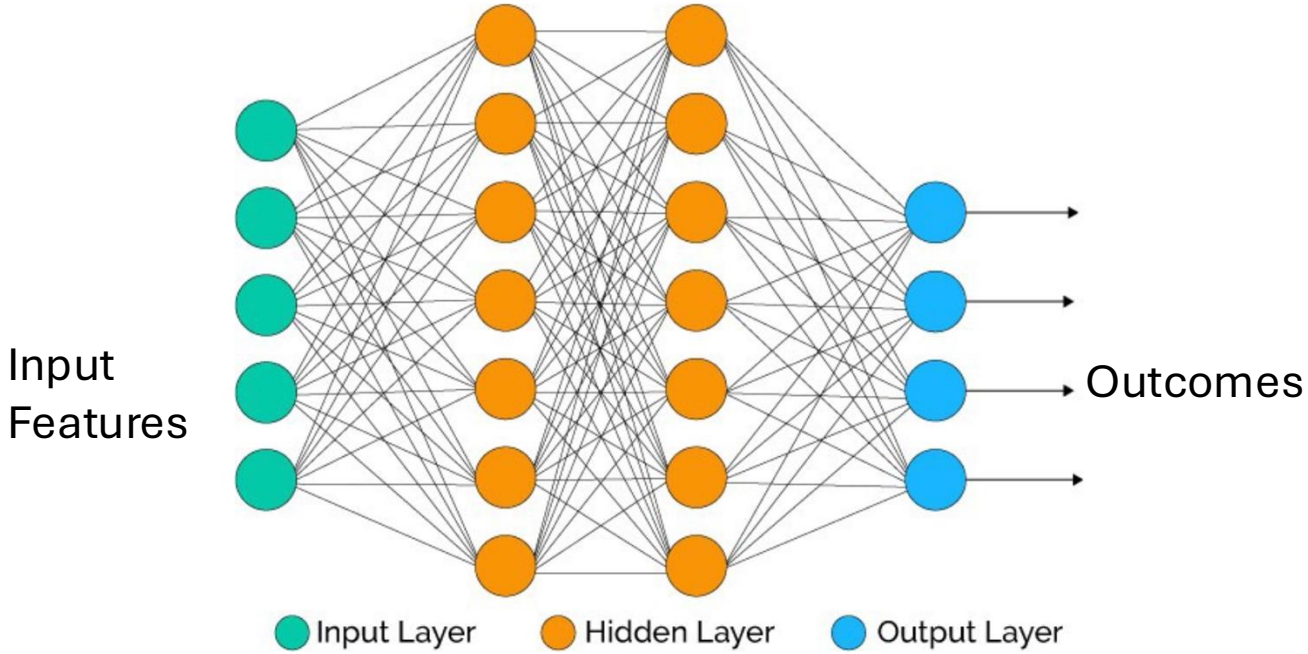
1. Build data nuggets :
2. Assign subgroup weights
3. Calculate estimands on each data nugget
4. Obtain the ATE weighted average of the data nuggets ATE's

Auto-encoder /Decoder for compressing data into a lower-dimensional "latent space"

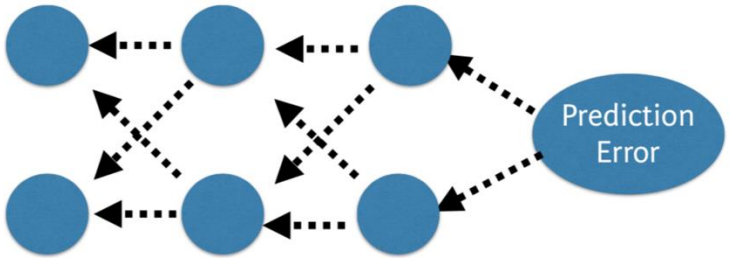
DEEP LEARNING: TENSOR FLOW. KERAS BACKEND

Multilayer Neural network. Runs using Python on many platforms

H2o: Another software that run on its one web server. Free access.

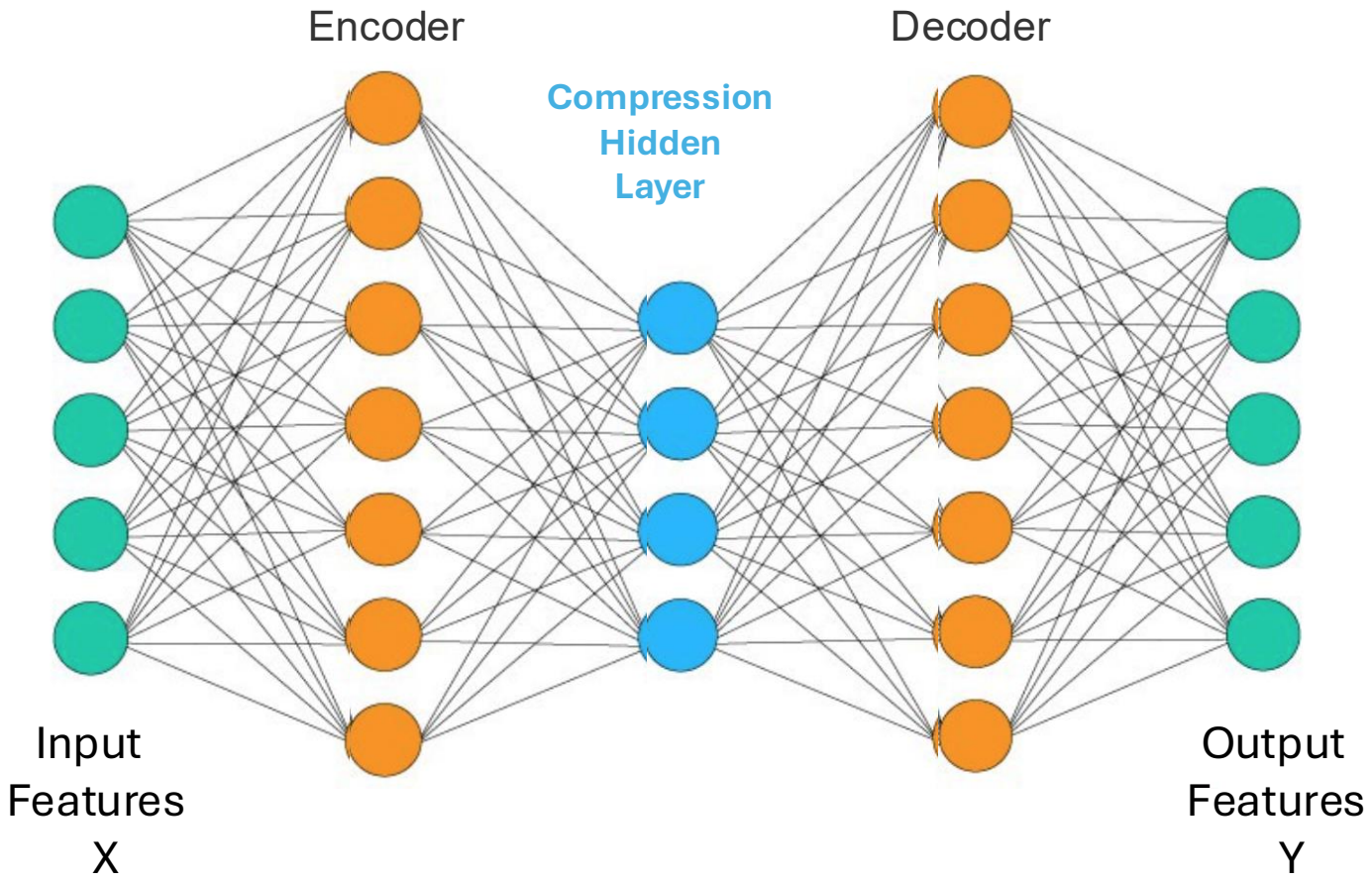


Fitting algorithm uses error backward propagation and thousands of epochs



https://bradleyboehmke.github.io/HOML/autoencoders.html

Auto-encoder /Decoder for compressing data into a lower-dimensional "latent space"



Autoencoders- decoders
Stacked Autoencoders- decoders

Auto-encoder /Decoder for compressing data into a lower-dimensional "latent space"

Autoencoder/decoder :: Nonlinear principal components. The compression hidden layer has less neurons so if makes the neural net to reduce the dimensionality of the data. It represents a method for dimension reduction that is comparable to Principal components but in a nonlinear way.

The algorithm minimizes the loss function

Loss Function: D(X,Y)

The features in the compressed hidden layer are also called Deep Features and are the equivalent to Principal components

Applications of VAE to matching and control augmentation

- Autoencoder decoder have a problem of scaling when dealing with binary/categorical/discrete variables.
- Suppose *X* a binary variable with 1% of 1's 99%0's *Y*=*X*/sd(X)

X=0	X=1	sd(X)	Y=X/sd(X)	FY	SD(FY)
99%	1%	0.0995	(0, 10.05)	(-0.031,2.23)	0.353
98%	2%	0.14	(0, 7.14)	(-0.038,2.16)	0.375
97%	3%	0.17	(0, 5.86)	(-0.044,2.09)	0.394
96%	4%	0.19	(0, 5.10)	(-0.050,2.04)	0.410

- Variational Autoencoder(VAE)/Decoder solves the problem by optimizing scores assigned to the binary/categorical values.
- This is similar to the categorical time series approach.

Other applications / Research in progress

These are on-going projects with my students and colleagues that are in development.

- Auto-encoder models for Augmenting Clinical Data from Real World data.

  With M. Cui, Z. Ke and G. Moran

  Faster Auto-encoder models replace the genetic algorithm

- Extending clinical evidence to real-world populations. (with Berhanu A.)

  Focus on intersection of R-W & CS and segment R-W outside of intersection

- A pseudo-randomization proposal for a clinical studies in distress (with V. Dragalin, S. Weigle)

  Clinical study: BL Randomization => Interim analysis => GA Randomization

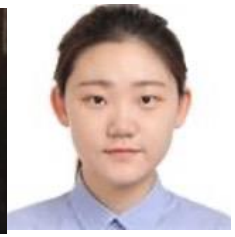
- Use of Natural Hermite and PE indices to a better implementations of training and testing and cross-validation methods.

- Randomization of non-clinical studies using the proposed indices. (S. Weigle, D. Sargsyan)

Collaborators

Collaborators:




Demissie Alemayehu Tray Beavers, Mingshi Cui Mahan Dastgiri, Yajie Duan, Zern Ke , G Moran, Davit Sargsyan, Sofia Weigle