DETECTION OF FALSE INVESTMENT STRATEGIES USING UNSUPERVISED LEARNING METHODS

Marcos López de Prado
Michael J. Lewis

First version: April 4, 2018
This version: November 1, 2018

Marcos López de Prado, Ph.D., is a principal and the head of machine learning at AQR Capital Management, in Greenwich, CT. He is also an adjunct professor at Cornell University, in Ithaca, NY. Corresponding author: mldp@quantresearch.org

Michael J. Lewis, Ph.D., is a vice president at Guggenheim Partners, in New York, NY.

DETECTION OF FALSE INVESTMENT STRATEGIES USING UNSUPERVISED LEARNING METHODS

ABSTRACT

In this paper we address the problem of selection bias under multiple testing in the context of investment strategies. We introduce an unsupervised learning algorithm that determines the number of effectively uncorrelated trials carried out in the context of a discovery. This estimate is critical for estimating the familywise false positive probability, and for filtering out false investment strategies.

Keywords: Backtest overfitting, selection bias, multiple testing, quantitative investments, machine learning.

JEL Classification: G0, G1, G2, G15, G24, E44.
AMS Classification: 91G10, 91G60, 91G70, 62C, 60E.

We wish to thank Prof. Germán G. Creamer and two anonymous referees for their help and useful comments.

1. INTRODUCTION

Finance lacks laboratories where experiments can be conducted while controlling for environmental conditions. For example, we cannot test the cause of the Flash Crash by reproducing the events of that date, while subtracting the trades of individual participants in order to derive a cause-effect mechanism. This elementary exercise, so common in Physics laboratories such as Berkeley Lab or CERN, is unavailable to financial researchers (López de Prado [2017]).

In absence of this essential scientific tool, financial researchers often resort to conducting backtests, which are simulations of how an investment portfolio would have performed under a particular historical scenario. The performance of such portfolio is often measured in terms of the Sharpe ratio (SR), which has become *de facto* the most popular investment performance metric. The distributional properties of the SR are well-known, allowing researchers to use this statistic to test the profitability of a strategy for a given confidence level (Lo [2002], Bailey and López de Prado [2012]).

A false positive occurs when a statistical test rejects a true null hypothesis. The probability of obtaining a false positive is set by the significance level (usually 5%). This false positive probability does not remain constant, and it necessarily increases as more than one test is conducted on the same data. The implication is that, applying the same rejection threshold for the null hypothesis under multiple testing will grossly underestimate the probability of obtaining a false positive. The practice of carrying out multiple tests without adjusting the rejection threshold is so widespread and misleading that the American Statistical Association considers it unethical (American Statistical Association [1997]). In particular, if we test multiple strategies on the same data, we should demand an increasing SR for the same false positive probability (Bailey et al. [2014a], Bailey and López de Prado [2014]).

Backtest overfitting occurs when a researcher makes a false discovery (finds a false positive) as a result of selecting the best outcome out of a multiplicity of backtests conducted on the same dataset. As soon as a researcher executes more than one backtest on a given dataset, backtest overfitting is taking place with a non-null probability. The goal of this paper is to provide a practical methodology that will allow researchers to compute and report the probability that an investment strategy is a false positive, while controlling for selection bias under multiple testing (SBuMT).

The rest of the paper is organized as follows: Section 2 reviews the literature. Section 3 lists the contributions made by this paper. Section 4 describes the distributional properties of the SR. Section 5 computes the probability that a strategy is a false positive in a single-test setting. Section 6 states the false strategy theorem. Section 7 explains how to compute the probability that a strategy is a false positive in a multiple-test setting. Section 8 provides practical solutions to the estimation of the probability of a false positive. Section 9 demonstrates empirically the accuracy of these practical solutions. Section 10 summarizes our findings.

2. REVIEW OF THE LITERATURE

In a series of papers, William Sharpe introduced the notion of measuring the performance of portfolio managers in terms of their risk-adjusted returns (Sharpe [1966, 1975, 1994]). This makes intuitive sense, since the portfolio that maximizes returns subject to a level of risk is a member of the Markowitz's efficient frontier. This performance measure quickly grew in popularity and became by far the premier statistic used to compare the performance across portfolio managers (Bailey and López de Prado [2012]).

Lo [2002] studied the distributional properties of the SR. He concluded that, under the assumption of independent and identically distributed (IID) Normal returns, the SR estimator follows a Normal distribution with mean SR and a standard deviation that depends on the very value of SR and the number of observations. Mertens [2002] found that the Normality assumption on returns could be dropped, and still the estimated SR would follow a Normal distribution. Christie [2005] derived a limiting distribution that only assumes stationary and ergodic returns, thus allowing for time-varying conditional volatilities, serial correlation and even non-IID returns. Surprisingly, Opdyke [2007] proved that the expressions in Mertens [2002] and Christie [2005] are in fact identical. Bailey and López de Prado [2012] derived the probability that the true SR exceeds a given benchmark level, under non-Normal returns.

Until 2014, all estimates of the SR assumed that returns were the result of a single trial. In a world where researchers routinely conduct millions of backtests, clearly that is an unrealistic assumption. To address this problem, Bailey and López de Prado [2014] introduced the *deflated Sharpe ratio* (DSR), which computes the probability that the true SR is positive while controlling for SBuMT.

In this paper we focus on Type I errors (false positives) rather than Type II errors (false negatives), because the former are actual economic losses, whereas the latter are opportunity losses. A hedge fund manager has a vested interest in minimizing Type II errors, while he receives a free call option on Type I errors. In other words, investors participate in the upside and downside, while managers participate only in the upside. Therefore, investors may adopt a "safety first" principle and concentrate on Type I errors, knowing that financial incentives take care of the Type II errors. For a treatment of both errors, see Harvey et al. [2018b] and López de Prado and Lewis [2018].

In general terms, the statistics literature on multiple testing works with two different definitions of Type I error: First, the Familywise Error Rate (FWER) is defined as the probability that *at least one* false positive takes place. FWER-based tests are designed to control for a single false positive (Holm [1979]). Second, the False Discovery Rate (FDR) is defined as the expected value of the ratio of false positives to predicted positives. FDR-based tests are designed to generate Type I errors at a constant rate, proportional to the number of predicted positives (Benjamini and Hochberg [1995], Benjamini and Liu [1999], Benjamini and Yekutieli [2001]). In most scientific and industrial applications, FWER is considered overly punitive, and authors prefer to use FDR. For example, it would be impractical to design a car model where we control for the probability that a single unit will be defective. However, in the context of finance, we advise against the use of FDR. The reason is, an investor does not typically allocate funds to all strategies with predicted positives within a family of trials, where a proportion of them are likely

to be false. Instead, investors are only introduced to the single best strategy out of a family of millions of alternatives. Following the car analogue, in finance there is actually a single car unit produced per model, which everyone will use. If the only produced unit is defective, everyone will crash. For example, investors are not exposed to the dozens of alternative model specifications tried by Fama and French. They have only been told about the one specification that Fama and French found to be best, and they have no ability to invest in their alternative models that passed individual statistical significance tests. Hence we argue that, in the context of financial applications, the more realistic Type I error definition is to control for a single error, not for an error rate. Accordingly, the procedure explained in this paper applies a FWER definition of Type I error.

3. OUR CONTRIBUTIONS

Bailey and López de Prado [2012, 2014] and Bailey et al. [2014] introduced the *False Strategy theorem* (see Section 6.1), and demonstrated how a SR estimate can be used to reject false discoveries under non-Normal returns while controlling for SBuMT. Critically, this theorem required the estimation of two meta-research variables, in the sense that they are variables related to the research process itself, rather than the outcome of the research. These two meta-research variables in question are: (1) The estimation of the number of effectively uncorrelated tests ($\mathrm{E}[K]$); and (2) the variance of the SR across the K effectively uncorrelated tests ($\mathrm{E}\left[\mathrm{V}[\{SR_k\}]\right]$). With the help of both variables, we can discount the likelihood of "lucky findings", that is, random patterns that appear naturally in the data but are meaningless. In this paper we provide practical solutions to the estimation of these two critical meta-research variables.

Important papers on this subject, published by Campbell Harvey and his coauthors, include Harvey et al. [2015, 2016, 2018a]. Their work shares similarities with ours, particularly as it relates to our concern that the practical totality of academic papers published in financial economics do not control for SBuMT, and the implication that most discoveries in empirical finance are likely to be false. Despite these similarities, our goals and mathematical approaches are different, as explained in Harvey et al. [2015, Section 3.4].

Particularly relevant is Harvey et al. [2015], which applies the Šidák correction (Šidák [1967]) to estimate the probability of observing a maximal SR that exceeds a given threshold. Their key assumptions are that returns are Normally distributed, and that trials are either independent or there is a constant average correlation between trials. Our method is different in three ways:

1. **We do not assume that returns follow a Normal distribution**. Empirical studies show that hedge fund returns exhibit substantial negative skewness and positive excess kurtosis. Wrongly assuming that returns are Normal underestimates the false positive probability (see Section 5). Our derived probability of a false discovery incorporates information regarding the trials' sample length, and the skewness and kurtosis of the observed returns.
2. **Our method is based on Extreme Value Theory, rather than Šidák's correction**. We derive the probability of a false positive adjusted for SBuMT through the direct application of the False Strategy theorem (see Section 6 herein). Notably, the False Strategy theorem uses the variance of the trials' SRs to accurately estimate the threshold

that the maximal SR must exceed to be statistically significant (see Section 6.2). Incorporating this variance information is critical when returns are not drawn from an IID Normal distribution (see Section 4).

3. **We do not assume a constant average correlation across trials**. A family of backtests often contains heterogeneous strategies. Trials that belong to the same strategy tend to be highly correlated among themselves, while trials that belong to different strategies tend to exhibit a lower correlation. This clustering of trials around heterogeneous strategies leads to a hierarchical structure, which can be highly irregular and complex. Assuming a constant correlation across all trials fails to recognize that hierarchical structure, biasing the estimates of the number of independent trials ($\mathrm{E}[K]$) and the false positive probability.

Generally speaking, our approach is data-intensive and closer to the machine learning literature, whereas Harvey et al.'s is closer to the econometrics literature. We advise readers to become familiar with both, as they can be seen as complementary. In particular, our unsupervised learning method for estimating the number of effectively uncorrelated tests ($\mathrm{E}[K]$) should be useful to both approaches.

4. THE NORMALITY OF THE SHARPE RATIO

Consider an investment strategy with excess returns (or risk premia) $\{r_t\}$, $t = 1, \ldots, T$, which follow an IID Normal distribution,

$$r_t \sim \mathcal{N}[\mu, \sigma^2]$$

where $\mathcal{N}[\mu, \sigma^2]$ represents a Normal distribution with mean μ and variance σ^2. The SR (non-annualized) of such strategy is defined as

$$SR = \frac{\mu}{\sigma}$$

Because parameters μ and σ are not known, SR is estimated as

$$\widehat{SR} = \frac{\mathrm{E}[\{r_t\}]}{\sqrt{\mathrm{V}[\{r_t\}]}}$$

Under the assumption that returns follow an IID Normal distribution, Lo [2002] derived the asymptotic distribution of $\widehat{SR}$ as

$$\left(\widehat{SR} - SR\right) \xrightarrow{a} \mathcal{N}\left[0, \frac{1 + \frac{1}{2}SR^2}{T}\right]$$

Equivalently, under the assumption that returns follow an IID Normal distribution, Harvey et al. [2015] transform $\widehat{SR}$ into a t-ratio, which follows a t-distribution with $T - 1$ degrees of freedom. In this paper we refrain from following that approach, as empirical evidence shows that hedge

fund strategies exhibit substantial negative skewness and positive excess kurtosis (among others, see Brooks and Kat [2002], Ingersoll et al. [2007]). Wrongly assuming that returns follow an IID Normal distribution can lead to a gross underestimation of the false positive probability.

Under the assumption that returns follow an IID non-Normal distribution, Mertens [2002] derived the asymptotic distribution of $\widehat{SR}$ as

$$\left(\widehat{SR} - SR\right) \xrightarrow{a} \mathcal{N}\left[0, \frac{1 + \frac{1}{2}SR^2 - \gamma_3 SR + \frac{\gamma_4 - 3}{4}SR^2}{T}\right]$$

where γ_3 is the skewness of $\{r_t\}$, and γ_4 is the kurtosis of $\{r_t\}$ ($\gamma_3 = 0$ and $\gamma_4 = 3$ when returns follow a Normal distribution). Shortly after, Christie [2005] and Opdyke [2007] discovered that, in fact, Mertens' equation is also valid under the more general assumption that returns are stationary and ergodic (not necessarily IID). The key implication is that $\widehat{SR}$ still follows a Normal distribution even if returns are non-Normal, however with a variance that partly depends on the skewness and kurtosis of the returns. In the next section we utilize this result to express the SR statistic in the probabilistic space. Such metric can be used directly to determine the probability that a discovery made after a single trial is a false positive.

5. THE PROBABILISTIC SHARPE RATIO

The probabilistic Sharpe ratio (PSR) provides an adjusted estimate of the SR, by removing the inflationary effect caused by short series with skewed and/or fat-tailed returns. Given a user-defined benchmark level SR^*, PSR estimates the probability that an observed $\widehat{SR}$ exceeds SR^*. Following Bailey and López de Prado [2012], PSR can be estimated as

$$\widehat{PSR}[SR^*] = Z\left[\frac{\left(\widehat{SR} - SR^*\right)\sqrt{T-1}}{\sqrt{1 - \hat{\gamma}_3\widehat{SR} + \frac{\hat{\gamma}_4 - 1}{4}\widehat{SR}^2}}\right]$$

where $Z[.]$ is the CDF of the standard Normal distribution, T is the number of observed returns, $\hat{\gamma}_3$ is the skewness of the returns, and $\hat{\gamma}_4$ is the kurtosis of the returns. Note that $\widehat{SR}$ is the non-annualized estimate of SR, computed on the same frequency as the T observations. For a given SR^*, $\widehat{PSR}$ increases with greater $\widehat{SR}$ (in the original sampling frequency, i.e. non-annualized), or longer track records (T), or positively skewed returns ($\hat{\gamma}_3$), but it decreases with fatter tails ($\hat{\gamma}_4$).

6. THE FALSE STRATEGY THEOREM

For the reader's convenience, in this section we will discuss the theorem needed to further adjust $\widehat{PSR}$ for the inflationary effect caused by SBuMT. A proof of this statement can be found in Bailey et al. [2014].

Given a sample of IID-Gaussian Sharpe ratios, $\{\widehat{SR}_k\}$, $k = 1, \ldots, K$, with $\widehat{SR}_k \sim \mathcal{N}\left[0, \mathrm{V}[\{\widehat{SR}_k\}]\right]$, then

$$\mathrm{E}\left[\max_k\{\widehat{SR}_k\}\right](\mathrm{V}[\{\widehat{SR}_k\}])^{-1/2} \approx (1-\gamma)Z^{-1}\left[1-\frac{1}{K}\right] + \gamma Z^{-1}\left[1-\frac{1}{Ke}\right]$$

where $Z^{-1}[.]$ is the inverse of the standard Gaussian CDF, e is Euler's number, and γ is the Euler-Mascheroni constant. The implication is that, unless $\max_k\{\widehat{SR}_k\} \gg \mathrm{E}[\max_k\{\widehat{SR}_k\}]$, the discovered strategy is likely to be a *false positive*. In Section 7 we will evaluate this likelihood.

7. THE DEFLATED SHARPE RATIO

In accordance with the previous result, we define the deflated Sharpe ratio (DSR) as the probability that the true SR exceeds a user-defined benchmark level SR^*, where that level is adjusted to reflect the multiplicity of trials. Following Bailey and López de Prado [2014], DSR can be estimated as $\widehat{PSR}[SR^*]$, where the benchmark SR (SR^*), is no longer user-defined. Instead, SR^* is estimated as

$$SR^* = \sqrt{\mathrm{V}[\{\widehat{SR}_k\}]}\left((1-\gamma)Z^{-1}\left[1-\frac{1}{K}\right] + \gamma Z^{-1}\left[1-\frac{1}{Ke}\right]\right)$$

where $\mathrm{V}[\{\widehat{SR}_k\}]$ is the variance across the trials' estimated SR, K is the number of independent trials, $Z[.]$ is the CDF of the standard Normal distribution, γ is the Euler-Mascheroni constant, and $k = 1, \ldots, K$.

The rationale behind DSR is the following: Given a set of SR estimates, $\{\widehat{SR}_k\}$, its expected maximum is greater than zero, even if the true SR is zero. Under the null hypothesis that the actual SR is zero, $H_0: SR = 0$, we know that the expected maximum SR can be estimated as the SR^*. Indeed, SR^* increases quickly as more independent trials are attempted (K), or the trials involve a greater variance $(\mathrm{V}[\{\widehat{SR}_k\}])$. In order to reject the null hypothesis that the strategy is uninformed ($H_0: SR = 0$), the observed SR ($\widehat{SR}$) must be statistically significantly greater than the expected SR after controlling for SBuMT (SR^*). Thus, DSR gives us the confidence level, that is, the probability complementary to the false positive rate. For example, in order to reject the null hypothesis, $H_0: SR = 0$, with a 5% significance level, the observed DSR must exceed 0.95.

8. PRACTICAL CONSIDERATIONS

In practice, the estimation of the false positive probability requires the evaluation of six variables:

1. $\widehat{SR}$
2. T

3. $\hat{\gamma}_3$
4. $\hat{\gamma}_4$
5. $\mathrm{E}[K]$
6. $\mathrm{E}\left[\mathrm{V}[\{\widehat{SR}_k\}]\right]$

Of these six variables, (1)-(4) are either directly observable or can be estimated from the selected strategy. However, (5)-(6) are meta-research variables, in the sense that they are intrinsic to the research process itself, and they cannot be estimated from the selected strategy.

There are two major reasons why (5)-(6) are usually unknown. First, it is common for researchers to hide, not track, not report or underreport (5)-(6). The motivations may vary, and they could range all the way between negligence and outright fraud. Regardless of the motivations, the implication is that ignorance of (5)-(6) makes it impossible to assess whether a discovery is false. Second, even those careful and knowledgeable researchers who track every single trial that takes place face the problem that trials are not usually *independent*. The number of independent trials K is less or equal to the number of trials N. In the following sections, we will show how (5)-(6) can be estimated in practice.

8.1. ESTIMATION OF THE NUMBER OF CLUSTERED TRIALS, E[K]

While finding independent trials may not be feasible, given that likely all strategies will be dependent to varying degrees, we consider clustering the strategies and using those clusters as a proxy. To that end, our goal is to develop an algorithm that, given N series, will partition them into an optimal number of K subgroups, or clusters. Ideally, each cluster will have high intra-cluster correlations and low inter-cluster correlations. We denote this algorithm ONC, since it searches for the *optimal number of clusters* within a correlation matrix.

Given that our goal is to cluster correlated strategies, we first assume that we have a correlation matrix ρ for our strategies, where ρ_{ij} is the correlation of the returns between strategies i and j. Next, we need a metric for clustering the strategies, specifically one where higher correlations map to smaller (closer) distances. For this, we consider the proper distance matrix D, where

$$D_{i,j} = \sqrt{\frac{1}{2}(1 - \rho_{ij})}$$

for $i, j = 1, \ldots, N$. This definition of distance is a proper metric in the sense that it satisfies the four classical axioms: Non-negativity, identity, symmetric and sub-additivity. Furthermore, we wish to consider a more global distance rather than local distance for improved clustering. Therefore, our clustering will be performed on the final Euclidean distance matrix $\widetilde{D}$ where

$$\widetilde{D}_{i,j} = \sqrt{\sum_k \left(D_{ik} - D_{jk}\right)^2}$$

In doing so, ONC works on a distance of distances ($\widetilde{D}$), rather than on a simple distance matrix (D). The reason is, while $D_{i,j}$ is a direct function of ρ_{ij} (a single correlation), $\widetilde{D}_{i,j}$ incorporates information about the entire system, thereby reducing noise and adding robustness to the procedure (López de Prado [2016a]).

With the above formed distance matrix $\widetilde{D}$, we next consider the clustering methodology. One possibility would be to use the K-means algorithm on our distance matrix $\widetilde{D}$. While K-means is simple and frequently effective, it does have two notable limitations: First, the algorithm requires a user-set number of clusters K, which is not necessarily optimal a priori. Second, the initialization is random, and hence the effectiveness of the algorithm is similarly random.

In order to address these two concerns, we need to modify the K-means algorithm. The first modification is to introduce an objective function, so that we can find the "optimal K." For this, we utilize the silhouette score introduced by Rousseeuw [1987]. As a reminder, for a given node i and a given clustering, the silhouette score S_i is defined as

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

where a_i is the average distance between i and all other nodes in the same cluster, and b_i is the smallest average distance between i and all the nodes in any other cluster. Effectively, this is a measure comparing intra-cluster distance and inter-cluster distance. A $S_i = 1$ means that node i is clustered well, while $S_i = -1$ means that i was clustered poorly. Our measure of quality q for a given clustering is thus set to

$$q = \frac{\mathrm{E}[\{S_i\}]}{\sqrt{\mathrm{V}[\{S_i\}]}}$$

The second modification deals with K-mean's initialization problem. At the base level, our clustering algorithm performs the following operation: First, we are given a NxN correlation matrix ρ, from which we evaluate the distance matrices D and $\widetilde{D}$. Second, we perform a double for…loop. In the first loop, we try different $k = 2, \dots, N-1$ on which to cluster via K-means for one given initialization, and evaluate the quality q for each clustering. The second loop repeats the first loop multiple times, thereby obtaining different initializations. Third, over these two loops, we select the clustering with the highest q. See Snippet 1 in the Appendix for an implementation of this operation in python.

The third modification to K-means deals with clusters of inconsistent quality. The base clustering may capture the more distinct clusters, while missing the less apparent ones. To address this issue, we evaluate the quality q_k of each cluster $k = 1, \dots, K$ given the clustering and silhouette scores obtained from the base clustering algorithm. We then take the average quality $\bar{q}$, and find the set of clusters with quality below average, $\{q_k | q_k < \bar{q}, k = 1, \dots, K\}$. Let us denote as K_1 the number of clusters in the set, $K_1 < K$. If the number of clusters to rerun is $K_1 \leq 2$, then we return

the clustering given by the base algorithm. However, if $K_1 > 2$, we rerun the clustering of the items in those K_1 clusters, while the rest are considered acceptably clustered.

We rerun the K_1 clusters in a recursive manner, rerunning the clustering on ρ, restricted to the nodes in the K_1 clusters. Doing so will return a, possibly new, optimal clustering for those nodes. To check its efficacy, we compare the average quality of the clusters to redo given the previous clustering to the average quality of the clusters given the new clustering. If the average quality improves for these clusters, we return the accepted clustering from the base clustering concatenated with the new clustering for the nodes redone. Otherwise, we return the clustering formed by the base algorithm. See Snippet 2 in the Appendix for an implementation of this operation in python. Exhibits 1 and 2 outline the structure of the ONC algorithm.

[EXHIBIT 1 HERE]
[EXHIBIT 2 HERE]

8.2. ESTIMATION OF THE VARIANCE OF CLUSTERED TRIALS, $\mathrm{E}\left[\mathrm{V}\left[\{\widehat{SR}_k\}\right]\right]$

Upon completion of the clustering above, ONC has successfully partitioned our N strategies into K groups, each of which is construed of highly correlated strategies. In this section, our goal is to utilize the clustering to reduce the N strategies to $K \ll N$ cluster-level strategies. Upon creation of these "cluster strategies," we derive our estimate $\mathrm{E}\left[\mathrm{V}\left[\{\widehat{SR}_k\}\right]\right]$ for each $k = 1, \ldots, K$.

For a given cluster k, the goal is to form an aggregate cluster returns time series $S_{k,t}$. This necessitates choosing a weighting scheme for the aggregation. We choose the minimum variance allocation, described in López de Prado [2016a], to mitigate the adverse effects of any strategies with larger variance. Let C_k denote the set of strategies in cluster k, $r_{i,t}$ the returns series for strategy i, Σ_k the covariance matrix restricted to strategies in C_k, and $w_{k,i}$, or w_k in vector notation, the weight for strategy $i \in C_k$. Then, we set

$$w_k = \frac{\Sigma_k^{-1} 1}{1' \Sigma_k^{-1} 1}$$

$$S_{k,t} = \sum_{i \in C_k} w_{k,i} \, r_{i,t}$$

where 1 is the characteristic vector of 1s. A robust method of computing w_k can be found in the Appendix. With the cluster returns time series $S_{k,t}$ now computed, we estimate each SR ($\widehat{SR}_k$). However, these $\widehat{SR}_k$ are not yet comparable, as their frequency of trading may vary. To make them comparable, we must first annualize each. Accordingly, we calculate the frequency of trading as

$$Years_k = \frac{Last\ Date_k - First\ Date_k}{365.25\ days}$$

$$Frequency_k = \frac{T_k}{Years_k}$$

where T_k is the length of the $S_{k,t}$, and $First\ Date_k$ and $Last\ Date_k$ are the first and last dates of trading for $S_{k,t}$, respectively. With this, we estimate the annualized Sharpe Ratio (aSR) as

$$\widehat{aSR}_k = \frac{\mathrm{E}\left[\{S_{k,t}\}\right] Frequency_k}{\sqrt{\mathrm{V}\left[\{S_{k,t}\}\right] Frequency_k}} = \widehat{SR}_k \sqrt{Frequency_k}$$

With these now comparable $\widehat{aSR}_k$, we can estimate the variance of clustered trials as

$$\mathrm{E}\left[\mathrm{V}\left[\{\widehat{SR}_k\}\right]\right] = \frac{\mathrm{V}\left[\{\widehat{aSR}_k\}\right]}{Frequency_{k^*}}$$

where $Frequency_{k^*}$ is the frequency of the selected strategy. We need to express the estimated variance of clustered trials in terms of the frequency of the selected strategy, in order to match the frequency of the $\widehat{SR}$ estimate used by the DSR equation (recall Sections 5 and 7). Otherwise, SR^* would not be estimated on the same frequency as $\widehat{SR}$.

9. EXPERIMENTAL VALIDATION OF E[K]

We now design a Monte Carlo experiment to verify the accuracy of the ONC algorithm introduced in Section 8.1. Our goal is to create a $N\ x\ N$ correlation matrix ρ from random data with a predefined number of blocks K, where ρ_{ij} is high inside a block and low outside the block. We can then verify that the ONC algorithm recovers the blocks we injected.

9.1. GENERATION OF RANDOM BLOCK CORRELATION MATRICES

First, given the tuple (N, M, K), we create a random block covariance matrix of size NxN, made up of K blocks, each of size $\geq M$. To do so, we randomly partition the N indices into K disjoint groups. Note that each block must be of size $M \geq 2$, as blocks of size 1 are difficult to identify as a cluster.

Let us describe the procedure for randomly partitioning N items into K groups, each of size at least M. First, note that this is equivalent to randomly partitioning $N' = N - K(M-1)$ items into K groups each of size at least 1, so we reduce our analysis to that. Next, consider randomly choosing $K-1$ distinct items, denoted as a set B, from the set $A = (1, \ldots, N'-1)$, then add N' to B, so that B is of size K. Thus, B contains $i_1, \ldots, i_K$, where $1 \leq i_1 < i_2 < \cdots < i_K = N'$. Given B, consider the K partition sets $C_1 = 0, \ldots, i_1 - 1$, $C_2 = i_1, \ldots, i_2 - 1$, …, and $C_K = i_{K-1}, \ldots, i_K - 1$. Given the i_j are distinct, each partition contains at least 1 element as desired, and furthermore completely partitions the set $(0, \ldots, N'-1)$. In doing so, each set C_j contains $i_j - i_{j-1}$ elements for $j = 1, \ldots, K$, letting $i_0 = 0$. We can generalize again by adding $M-1$ elements to each block.

Let each block $k = 1, \ldots, K$ have size x_k by x_k, where $x_k \geq M$, thus implying $x_1 + \cdots + x_K = N \geq MK$. First, for each block k, we create a time series S of length T that is made from IID standard Gaussians, then make copies of that to each column of a matrix X of size (T, x_k). Second, we add to each X_{ij} a random Gaussian with standard deviation $\sigma > 0$. By design, the columns of X will be highly correlated for small σ, and less correlated for large σ. Third, we evaluate the covariance matrix Σ_X for the columns of X, and add Σ_X as a block to Σ. Fourth, we add to Σ another covariance matrix with one block but larger σ. Finally, we derive the correlation matrix ρ related to Σ.

By design, ρ will have K blocks with high correlations inside each block, and low correlations otherwise. Exhibit 3 is an example of a correlation matrix constructed this way. See Snippet 3 in the Appendix for an implementation of this operation in python.

[EXHIBIT 3 HERE]

9.2. EXTRACTION OF E[K]

Using the above described procedure to create random $N \; x \; N$ correlation matrices with K blocks of size at least M, we test the efficacy of the ONC algorithm. For our simulations, we chose $N = 20, 40, 80, 160$. We set $M = 2$, and thus necessarily $\frac{K}{N} \leq \frac{1}{2}$. For each N, we test $K = 3, 6, \ldots$, up to $\frac{N}{2}$. Finally, we test 1000 random generations for each of these parameter sets.

Exhibit 4 displays various boxplots for these simulations. In particular, for $\frac{K}{N}$ in a given bucket, we display the boxplot of the ratio of K predicted by the clustering (denoted E[K]) to the actual K tested. Ideally, this ratio should be near 1. We observe that this clustering is very effective, frequently obtaining the correct number of clusters, with some outliers.

[EXHIBIT 4 HERE]

As a reminder, in a boxplot, the central box has the bottom set to the 25% percentile of the data (Q1), while the top is set to the 75% percentile (Q3). The interquartile range (IQR) is set to Q3-Q1. The median is displayed as a line inside the box. The "whiskers" extend to the largest datum less than $Q3 + 1.5IQR$, and the smallest datum greater than $Q1 - 1.5IQR$. All points outside that range are considered outliers.

10. CONCLUSIONS

In this paper we apply the False Strategy theorem, first proved in Bailey et al. [2014], to the prevention of false positives in finance. This requires the estimation of two meta-research variables that allow us to discount for the likelihood of "lucky findings." We estimate these two meta-research variables with the help of the ONC algorithm.

In particular, ONC extracts from a series of backtests the number of effectively uncorrelated trials. This number is useful in two applications: a) Estimating the expected value of the maximum Sharpe ratio, via the False Strategy Theorem (see Bailey and López de Prado [2014]

for an example); and b) deriving the FWER, via the Šidák correction (see Harvey and Liu [2015] for an example). Monte Carlo experiments demonstrate the precision of this method.

We think that ONC has multiple uses in finance. Many investing problems involve the extraction of an unknown number of clusters. For example, ONC could be used to identify the optimal number of economic sectors from a risk perspective. Risk parity investors could then allocate assets in a more diversified way, where the peer groups are not set in advance. More generally, ONC could be useful in situations where researchers are interested in finding the most uncorrelated groups without a change of basis (like in principal components analysis, PCA). This could be particularly helpful in addressing multicollinearity problems, where the standard PCA solution forces researchers to work with variables removed of economic intuition.

APPENDIX

A.1. THE BASE CLUSTERING ALGORITHM

The purpose of this step is to perform a first-pass estimate of $\mathrm{E}[K]$. First, we transform the correlation matrix into a distance matrix. On this distance matrix, we apply the K-means algorithm on alternative target number of clusters. For each target number of clusters, we perform a stochastic optimization, repeating the clustering operation `n_init` times. Among all the clustering alternatives, we choose the solution that achieves the highest quality score, defined as the t-value of the silhouette scores.

```
import numpy as np,pandas as pd
#-------------------------------------------------------------------------------
def clusterKMeansBase(corr0,maxNumClusters=10,n_init=10):
    from sklearn.cluster import KMeans
    from sklearn.metrics import silhouette_samples
    dist,silh=((1-corr0.fillna(0))/2.)**.5,pd.Series() # distance matrix
    for init in range(n_init):
        for i in xrange(2,maxNumClusters+1): # find optimal num clusters
            kmeans_=KMeans(n_clusters=i,n_jobs=1,n_init=1)
            kmeans_=kmeans_.fit(dist)
            silh_=silhouette_samples(dist,kmeans_.labels_)
            stat=(silh_.mean()/silh_.std(),silh.mean()/silh.std())
            if np.isnan(stat[1]) or stat[0]>stat[1]:
                silh,kmeans=silh_,kmeans_
    n_clusters = len( np.unique( kmeans.labels_ ) )
    newIdx=np.argsort(kmeans.labels_)
    corr1=corr0.iloc[newIdx] # reorder rows
    corr1=corr1.iloc[:,newIdx] # reorder columns
    clstrs={i:corr0.columns[np.where(kmeans.labels_==i)[0] ].tolist() for \
            i in np.unique(kmeans.labels_) } # cluster members
    silh=pd.Series(silh,index=dist.index)
    return corr1,clstrs,silh
```

Snippet 1 – Base Clustering

A.2. THE TOP-LEVEL CLUSTERING ALGORITHM

The purpose of this step is to perform a second-pass estimate of $\mathrm{E}[K]$. We evaluate the quality score for each cluster within the first-pass solution. Those clusters with quality greater or equal than average remain unchanged. We re-run the base clustering on clusters with below-average quality. The outputs of these re-runs are preserved only if their cluster quality improves.

```
#-------------------------------------------------------------------------------
def makeNewOutputs(corr0,clstrs,clstrs2):
    from sklearn.metrics import silhouette_samples
    clstrsNew,newIdx={},[]
    for i in clstrs.keys():
        clstrsNew[len(clstrsNew.keys())]=list(clstrs[i])
    for i in clstrs2.keys():
        clstrsNew[len(clstrsNew.keys())]=list(clstrs2[i])
    map(newIdx.extend, clstrsNew.values())
    corrNew=corr0.loc[newIdx,newIdx]
```

```
    dist=((1-corr0.fillna(0))/2.)**.5
    kmeans_labels=np.zeros(len(dist.columns))
    for i in clstrsNew.keys():
        idxs=[dist.index.get_loc(k) for k in clstrsNew[i]]
        kmeans_labels[idxs]=i
    silhNew=pd.Series(silhouette_samples(dist,kmeans_labels),index=dist.index)
    return corrNew,clstrsNew,silhNew
#-------------------------------------------------------------------------------
def clusterKMeansTop(corr0,maxNumClusters=10,n_init=10):
    corr1,clstrs,silh=clusterKMeansBase(corr0,maxNumClusters=corr0.shape[1]-1,n_init=n_init)
    clusterTstats={i:np.mean(silh[clstrs[i]])/np.std(silh[clstrs[i]]) for i in clstrs.keys()}
    tStatMean=np.mean(clusterTstats.values())
    redoClusters=[i for i in clusterTstats.keys() if clusterTstats[i]<tStatMean]
    if len(redoClusters)<=2:
        return corr1,clstrs,silh
    else:
        keysRedo=[];map(keysRedo.extend,[clstrs[i] for i in redoClusters])
        corrTmp=corr0.loc[keysRedo,keysRedo]
        meanRedoTstat=np.mean([clusterTstats[i] for i in redoClusters])
        corr2,clstrs2,silh2=clusterKMeansTop(corrTmp, \
            maxNumClusters=corrTmp.shape[1]-1,n_init=n_init)
        # Make new outputs, if necessary
        corrNew,clstrsNew,silhNew=makeNewOutputs(corr0, \
            {i:clstrs[i] for i in clstrs.keys() if i not in redoClusters},clstrs2)
        newTstatMean=np.mean([np.mean(silhNew[clstrsNew[i]])/np.std(silhNew[clstrsNew[i]]) \
            for i in clstrsNew.keys()])
        if newTstatMean<=meanRedoTstat:
            return corr1,clstrs,silh
        else:
            return corrNew,clstrsNew,silhNew
```

Snippet 2 – Top Level of Clustering

A.3. RANDOM CORRELATION BLOCK-MATRICES

In this section we present an algorithm for the generation of random correlation block-matrices, with a pre-determined number of clusters. After generating these matrices, we can shuffle their rows (and columns), and apply the ONC algorithm. We can repeat this process thousands of times to evaluate ONC's performance, while controlling for the matrix size and the number of clusters.

```
import numpy as np,pandas as pd
from scipy.linalg import block_diag
from sklearn.utils import check_random_state
#-------------------------------------------------------------------------------
def cov2corr(cov):
    # Derive the correlation matrix from a covariance matrix
    std=np.sqrt(np.diag(cov))
    corr=cov/np.outer(std,std)
    corr[corr<-1],corr[corr>1]=-1,1 # numerical error
    return corr
#-------------------------------------------------------------------------------
def getCovSub(nObs,nCols,sigma,random_state=None):
    # Sub correl matrix
    rng = check_random_state(random_state)
    if nCols==1:return np.ones((1,1))
```

```
    ar0=rng.normal(size=(nObs,1))
    ar0=np.repeat(ar0,nCols,axis=1)
    ar0+=rng.normal(scale=sigma,size=ar0.shape)
    ar0=np.cov(ar0,rowvar=False)
    return ar0
#-------------------------------------------------------------------------------
def getRndBlockCov(nCols,nBlocks,minBlockSize=1,sigma=1.,random_state=None):
    # Generate a random correlation matrix with a given number of blocks
    rng = check_random_state(random_state)
    parts=rng.choice(range(1,nCols-(minBlockSize-1)*nBlocks),nBlocks-1,replace=False)
    parts.sort()
    parts=np.append(parts,nCols-(minBlockSize-1)*nBlocks)
    parts=np.append(parts[0],np.diff( parts )) - 1 + minBlockSize
    cov=None
    for nCols_ in parts:
        cov_=getCovSub(int(max(nCols_*(nCols_+1)/2.,100)),nCols_,sigma,random_state=rng)
        if cov is None:cov=cov_.copy()
        else:cov=block_diag(cov,cov_)
    return cov
#-------------------------------------------------------------------------------
def randomBlockCorr(nCols,nBlocks,random_state=None,minBlockSize=1):
    # Form block covar
    rng = check_random_state(random_state)
    cov0=getRndBlockCov(nCols,nBlocks,minBlockSize=minBlockSize,\
                        sigma=.5,random_state=rng) # perfect block corr
    cov1=getRndBlockCov(nCols,1,minBlockSize=minBlockSize,\
                        sigma=1.,random_state=rng) # add noise
    cov0+=cov1
    corr0=cov2corr(cov0)
    corr0=pd.DataFrame(corr0)
    return corr0
```

Snippet 3 – Random block correlation matrix creation

A.4. MINIMUM VARIANCE ALLOCATION

In section 8.2, we wish to evaluate the minimum variance allocation for the strategies within a cluster k of size N_k. Note that the intra-cluster correlations will be high by design, and thus Σ_k may be ill-conditioned and difficult to invert. In practice, one could choose to approximate the weights by setting $w_{k,i}$ proportional to $\frac{1}{\sigma_i^2}$ as is typically done in inverse variance allocations. If more accuracy is desired, consider the following approximation. Let ρ be the average off-diagonal correlation in the correlation matrix for the cluster. Then, the covariance matrix is approximately

$$\Sigma_{\mathrm{k}} \approx \Sigma_{approx} = \begin{pmatrix} \sigma_1^2 & \cdots & \rho\sigma_1\sigma_{N_k} \\ \vdots & \ddots & \vdots \\ \rho\sigma_1\sigma_{N_k} & \cdots & \sigma_{N_k}^2 \end{pmatrix} = \rho\sigma\sigma^T + (1-\rho)\begin{pmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{N_k}^2 \end{pmatrix}$$

where

$$\sigma = \begin{pmatrix} \sigma_1 \\ \vdots \\ \sigma_{N_k} \end{pmatrix}$$

This is a rank one update. If our goal is to take the inverse of Σ_{approx}, we can utilize the Sherman-Morrison (SM) formula. Using the notation

$$\frac{1}{\sigma} = \begin{pmatrix} \frac{1}{\sigma_1} \\ \vdots \\ \frac{1}{\sigma_{N_k}} \end{pmatrix}$$

then the SM formula gives us

$$\Sigma_{\text{approx}}^{-1} = \frac{1}{1-\rho}\begin{pmatrix} \frac{1}{\sigma_1^2} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sigma_{N_k}^2} \end{pmatrix} - \frac{\rho}{(1-\rho)(1+(N_k-1)\rho)}\left(\frac{1}{\sigma}\right)\left(\frac{1}{\sigma}\right)^T$$

When computing the weights allocation, we are trying to evaluate $\Sigma_{\text{approx}}^{-1}1$. In this case, we find that

$$w_{k,i} \sim \frac{1}{\sigma_i^2} - \frac{\rho \sum_{j \in C_k} \frac{1}{\sigma_j}}{(1+(N_k-1)\rho)\sigma_i}$$

As is readily observable, if $\rho = 0$, this reduces to the standard inverse variance allocation. Snippet 4 implements this procedure in python.

```
import numpy as np,pandas as pd
#-------------------------------------------------------------------------------
def getIVP(cov,use_extended_terms=False):
    # Compute the minimum-variance portfolio
    ivp=1./np.diag(cov)
    if use_extended_terms:
        n=float(cov.shape[0])
        corr=cov2corr(cov)
        # Obtain average off-diagonal correlation
        rho=(np.sum(np.sum(corr))-n)/(n**2-n)
        invSigma=np.sqrt(ivp)
        ivp-=rho*invSigma*np.sum(invSigma)/(1.+(n-1)*rho)
    ivp/=ivp.sum()
    return ivp
```

Snippet 4 – Obtain minimum variance portfolio

REFERENCES

American Statistical Association (2016): "Ethical guidelines for statistical practice." Committee on Professional Ethics. Available at http://www.amstat.org/asa/files/pdfs/EthicalGuidelines.pdf

Bailey, D., J. Borwein, M. López de Prado, and J. Zhu (2014a): "Pseudo-mathematics and financial charlatanism: The effects of backtest overfitting on out-of-sample performance." *Notices of the American Mathematical Society*, Vol. 61, No. 5, pp. 458–471. Available at http://ssrn.com/abstract=2308659

Bailey, D., J. Borwein, M. López de Prado, and J. Zhu (2017): "The Probability of Backtest Overfitting." *Journal of Computational Finance*, Vol. 20, No. 4, pp. 39-70. Available at http://ssrn.com/abstract=2326253

Bailey, D. and M. López de Prado (2012): "The Sharpe ratio efficient frontier." *Journal of Risk*, Vol. 15, No. 2, pp. 3–44.

Bailey, D. and M. López de Prado (2014): "The deflated Sharpe ratio: Correcting for selection bias, backtest overfitting and non-normality." *Journal of Portfolio Management*, Vol. 40, No. 5, pp. 94-107.

Benjamini, Y., and Y. Hochberg (1995): "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society*, Series B, Vol. 57, pp. 289-300.

Benjamini, Y., and W. Liu (1999): "A Step-down Multiple Hypotheses Testing Procedure that Controls the False Discovery Rate under Independence." *Journal of Statistical Planning and Inference*, Vol. 82, pp. 163-170.

Benjamini, Y., and D. Yekutieli (2001): "The Control of the False Discovery Rate in Multiple Testing under Dependency." *Annals of Statistics*, Vol. 29, pp. 1165-1188.

Brooks, C., H. Kat (2002): "The Statistical Properties of Hedge Fund Index Returns and Their Implications for Investors." *Journal of Alternative Investments*, Vol. 5, No. 2 (Fall), pp. 26-44.

Christie, S. (2005): "Is the Sharpe Ratio Useful in Asset Allocation?" MAFC Research Paper No. 31, Applied Finance Centre, Macquarie University.

Harvey, C. and Y. Liu (2015): "Backtesting." *The Journal of Portfolio Management*, 42(1), pp. 13-28.

Harvey, C., Y. Liu and C. Zhu (2016): "…and the Cross-Section of Expected Returns." *Review of Financial Studies*, 29(1), pp. 5-68. Available at https://ssrn.com/abstract=2249314

Harvey, C. and Y. Liu (2018a): "Lucky Factors." Working paper. Available at https://ssrn.com/abstract=2528780

Harvey, C. and Y. Liu (2018b): "False (and Missed) Discoveries in Financial Economics." Working paper. Available at https://ssrn.com/abstract=3073799

Holm, S. (1979): "A Simple Sequentially Rejective Multiple Test Procedure." *Scandinavian Journal of Statistics*, Vol. 6, pp. 65-70.

Ingersoll, J., M. Spiegel, W. Goetzmann, I. Welch (2007): "Portfolio performance manipulation and manipulation-proof performance measures." *The Review of Financial Studies*, Vol. 20, No. 5, pp. 1504-1546.

Lo, A. (2002): "The Statistics of Sharpe Ratios." *Financial Analysts Journal* (July), pp. 36-52.

López de Prado, M. (2016a): "Building Diversified Portfolios that Outperform Out-of-Sample." *Journal of Portfolio Management*, Vol. 42, No. 4, pp. 59-69.

López de Prado, M. (2016b): "Mathematics and Economics: A reality check." *Journal of Portfolio Management*, Vol. 43, No. 1, pp. 5-8.

López de Prado, M. (2017): "Finance as an Industrial Science." *Journal of Portfolio Management*, Vol. 43, No. 4, pp. 5-9.

López de Prado, M. (2018a): Advances in Financial Machine Learning. 1st edition, Wiley. https://www.amazon.com/dp/1119482089

López de Prado, M. and M. Lewis (2018): "What is the optimal significance level for investment strategies?" Working paper. Available at https://ssrn.com/abstract=3193697

Mertens, E. (2002): "Variance of the IID estimator in Lo (2002)." Working paper, University of Basel.

Opdyke, J. (2007): "Comparing Sharpe ratios: So where are the p-values?" *Journal of Asset Management*, Vol. 8, No. 5, pp. 308–336.

Rousseeuw, P. (1987): "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Computational and Applied Mathematics*, Vol. 20, pp. 53–65.

Sharpe, W. (1966): "Mutual Fund Performance." *Journal of Business*, Vol. 39, No. 1, pp. 119–138.

Sharpe, W. (1975): "Adjusting for Risk in Portfolio Performance Measurement." *Journal of Portfolio Management*, Vol. 1, No. 2, Winter, pp. 29-34.

Sharpe, W. (1994): "The Sharpe ratio." *Journal of Portfolio Management*, Vol. 21, No. 1, Fall, pp. 49-58.

Šidák, Z. (1967): "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions." *Journal of the American Statistical Association*, Vol. 62, No. 318, pp. 626–633.

EXHIBITS

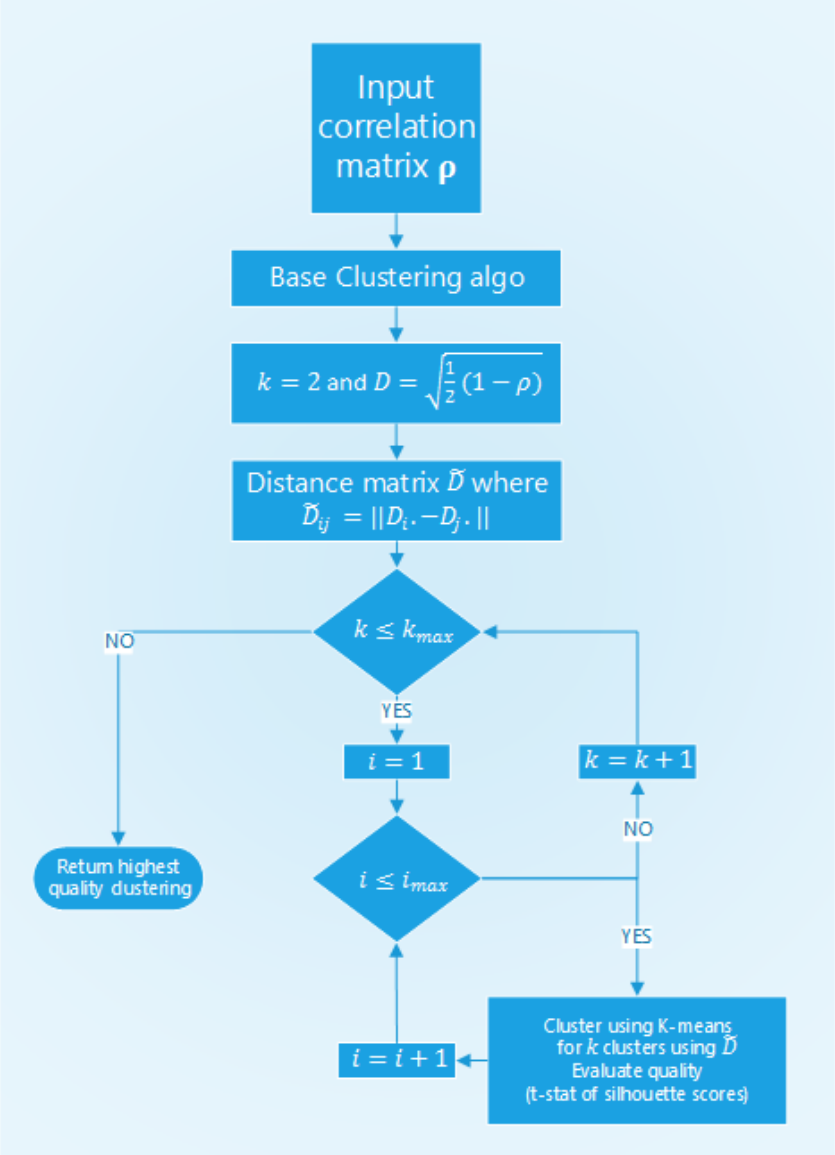

Exhibit 1 – Structure of ONC's base clustering stage

This exhibit outlines the workflow within ONC's base algorithm, highlighting the three ways in which it departs from the K-means algorithm: 1) The clustering is done on a distance of distances ($\tilde{D}$), rather than on the distance matrix (D); 2) it optimizes the Silhouette score; 3) it tries alternative initialization points to avoid local optima

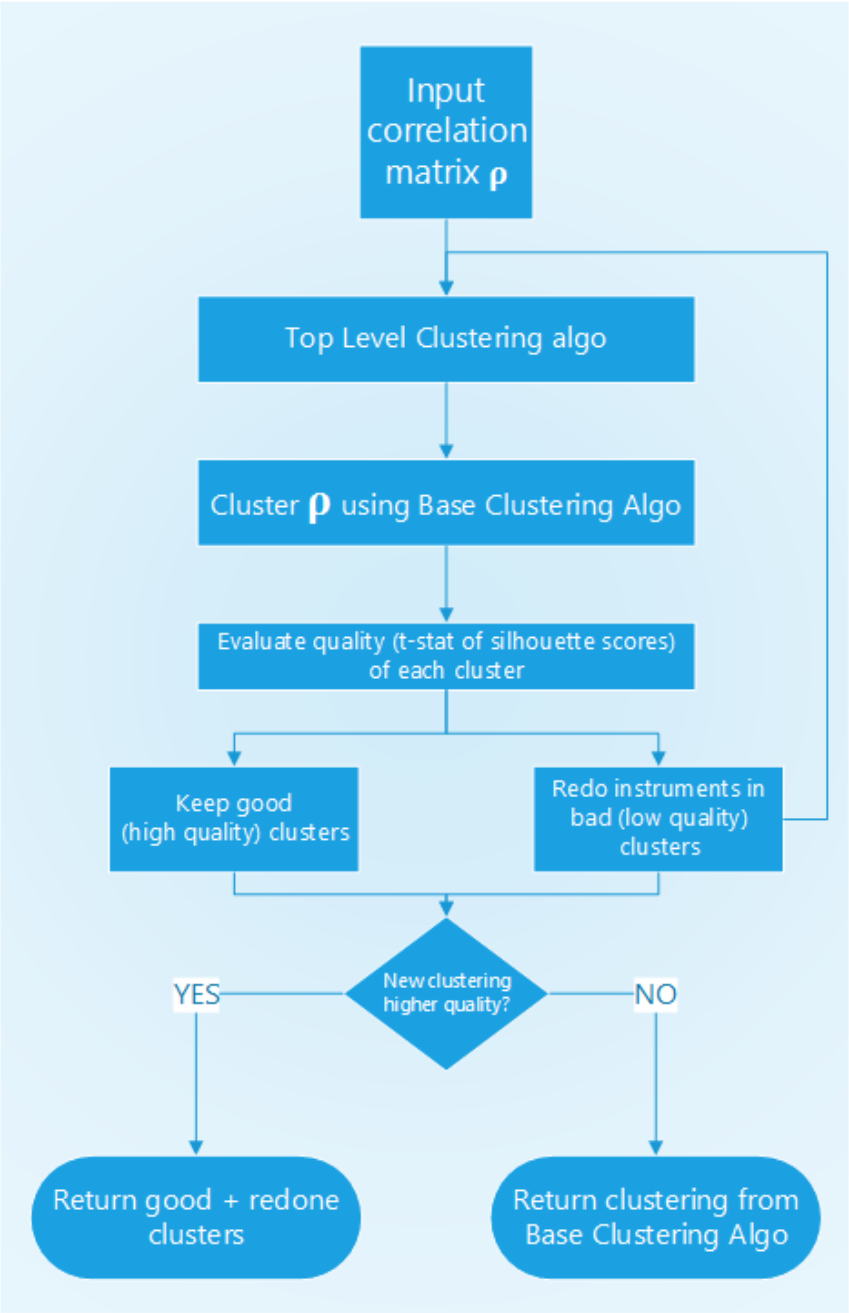

Exhibit 2 – Structure of ONC's higher-level stage

This exhibit outlines ONC's higher-level clustering, which seeks to reduce discrepancies across clusters quality.

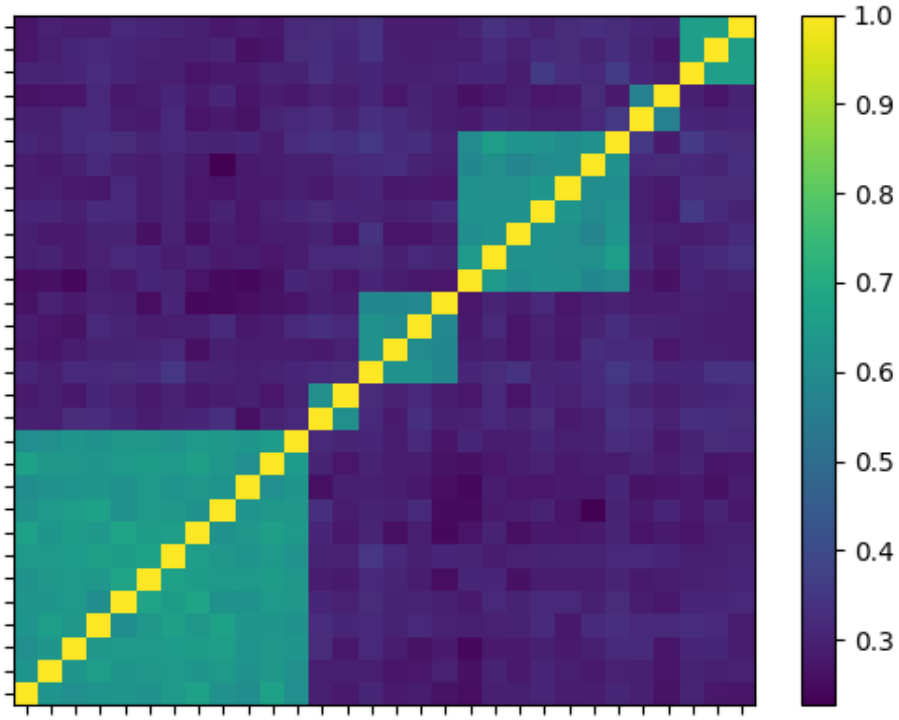

Exhibit 3 – Example of a random block correlation matrix

This exhibit plots a random block correlation matrix, generated using the method explained in section 9.1. Light colors indicate a high correlation, and dark colors indicate a low correlation. In this example, the number of blocks $K = 6$, each of varying size, with a total of $N = 30$ instruments.

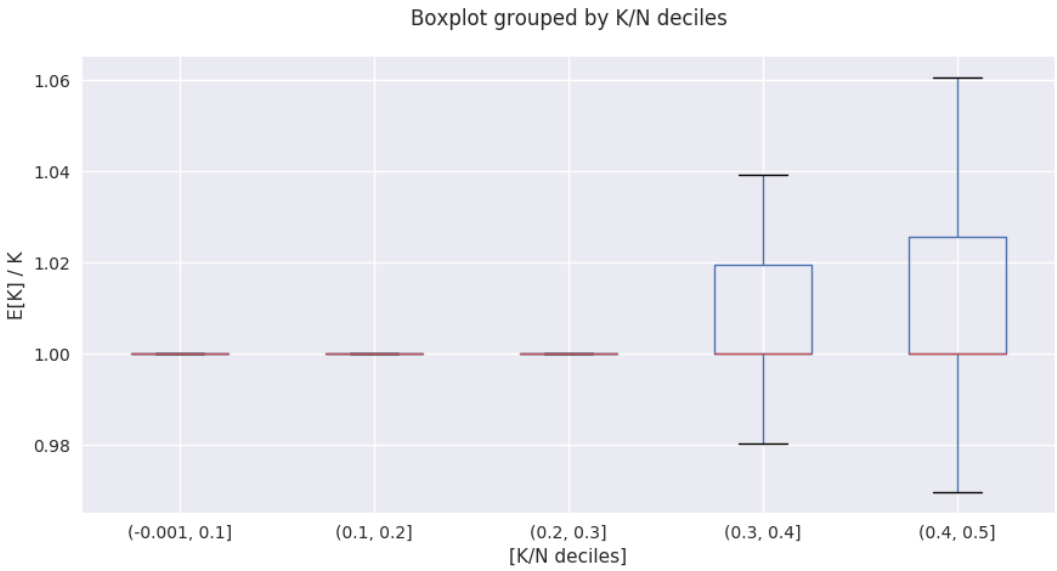

Exhibit 4 – Boxplots of estimated K / actual K for bucketed $\frac{K}{N}$

This exhibit plots the ratio between the extracted number of clusters (E[K]) and the actual number of clusters (K), for various deciles of K/N, where NxN is the size of the correlation matrix. These results were obtained from numerous random simulations across a variety of matrix sizes and cluster counts, namely $N = 20, 40, 80, 160$ and $K = 3, 6, ...$, up to $\frac{N}{2}$.The ONC algorithm provides an accurate estimation of K across all ratios of clusters per variable.