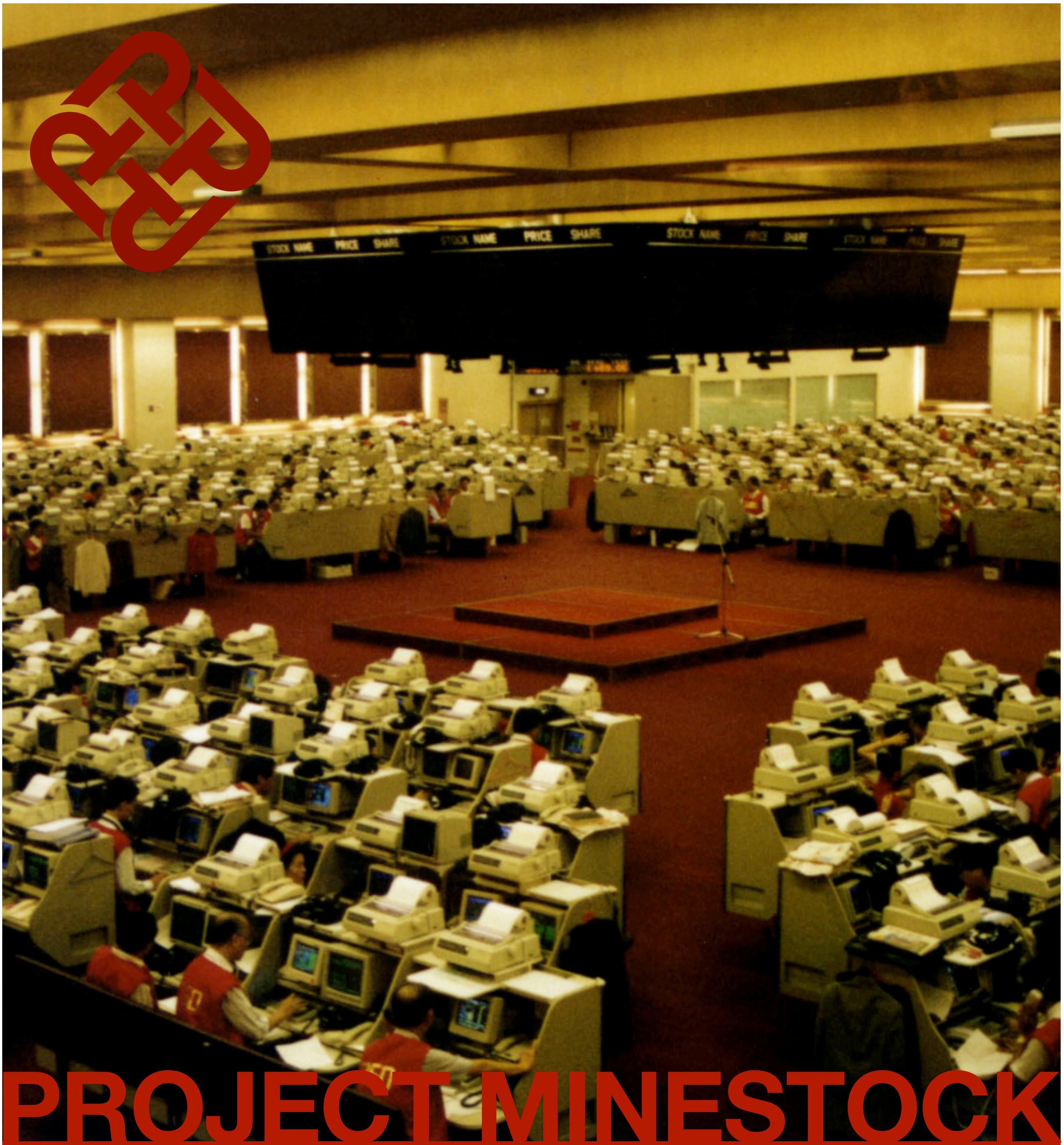


PROJECT MINESTOCK

Final Report

Written by TONG Man Kin (08502621D)
Supervised by Dr. CHAN Chun Chung Keith,
Co-examined by Dr. ZHANG Lei and Dr. LO Chi Lik Eric

BSc (Hons.) Major in Computing (61031-ASC)

April 14, 2011

Project MineStock
-
Final Report

Written by TONG Man Kin (08502621d)

Supervised by Dr. CHAN Chun Chung Keith
Co-examined by Dr. ZHANG Lei and Dr. LO Chi Lik Eric,
The Hong Kong Polytechnic University

April 14, 2011

Contents

List of Figures

List of Tables

1 Abstract

The financial tsunami in 2008 has turned the world into a turmoil. Its destructive effect has lasted until today, causing people not only to reevaluate the risk of their derivative investments, but also reconsider their current investment strategy. One important lesson we learned from the collapse of Lehman Brothers is that the derivative instruments are only effective in transferring a type of risk (i.e. market risk) to another (i.e. counter party risk). They have no use, ironically, to reduce the total amount of risks that one investor is exposing in, which is very different from what we usually perceived. Clearly, it would be much wiser for an individual investor to adopt the classical technique - portfolio diversification, instead of putting a vain hope, again, on those derivatives.

The writer believes a sophisticated portfolio optimization tool would satisfy the need of investors nowadays and help the promotion of rational investing. This tool, namely the 'MineStock Workbench', is therefore proposed to be built. By applying certain data mining techniques, particularly the clustering algorithms, assisted by theories of mathematical finance, this scientific tool is considered able to suggest both the optimal stock selection and their optimal weighting in portfolios, and thus the investors are able to diversify the risk of their investments.

Besides the implementation of portfolio optimization tool, this project also tried to evaluate the performance of the clustering algorithms which are chosen to apply. In the writer's studies, the 2 algorithms used have also outperformed the classical k-means technique for time series data.

In this report, the writer will firstly introduce the methodologies of the project, discuss how the algorithms and theories are expected to be able to help investors in their stock selection and portfolio optimization processes, secondly, outline the important details in the architecture and design of the system, and thirdly, introduce the way to use the system by going through the interfaces. Then the evaluation results will be discussed. Finally, the report will end by suggesting future works and conclusion.

2 Acknowledgment

I would like to express my sincere gratitude to my project supervisor, Prof. Chan Chun Chung Keith, who has spent so much time to discuss the project with me and gave me valuable guidance and suggestions. He is always patience to listen to my view, analyze my findings and lead me to the right directions.

My special thanks to Mr Franklin Leung from Department of Computing, Mr Raymond Chan and Mr Lawrence Fung from the School of Accounting and Finance, who provided me a good foundation on finance and investment, especially on the area of portfolio management.

Last but not least, I would like to acknowledge special thanks to my family and friends who gave me lots of encouragement and help. Finally, I am very grateful for my girl friend, Cheryl Tang, for her love and support during this year.

3 Background and Problems

This project aims at designing and developing a technique that is capable of discovering patterns in time series data. It also emphasizes on applications to ensure that the techniques developed can be used in finance for stock data analysis. After reviewing existing analysis techniques of stock data and understanding the investment behavior of individual investors, the author believes there is a lack of sophisticated portfolio optimization tools from them. Therefore, implementing a software tool which makes use of time series data mining techniques, and provides suggestions to users such that portfolios could be built with better diversification effect, is proposed for this project.

3.1 Failure of Derivatives

With a open and well-developed equity market, Hong Kong is flooded with a wide spectrum of derivative products, including different types of Equity-Linked Instruments (ELI), warrants, Callable Bull or Bear Contracts (CBBCs) and accumulators. Although accumulators are mainly traded by enterprises, other products have already been very common among individual investors. According to HKEx [1, 2], the percentage of warrants and CBBCs investors among adult population in Hong Kong has been greatly grown from 1% in 2000, to more than 12% in 2009. These products are often considered as useful instruments to hedge the risk of ordinary stocks, but seldom realized by individuals that they actually carry an even more complicated risk structure. Investors may suffer from total loss of capital by not knowing the characteristics of those derivatives, which is agonizing.

In the year 2008, the financial tsunami severely hit United States' economy, and the bankruptcy of a hundred-year financial institution, Lehman Brothers, spread its disastrous shock-wave around the world. Different kinds of derivatives issued by Lehman Brothers, turned out to be scrap papers overnight, along with the collapse of the company. Hong Kong, as one of the major market of these products, her investors lose all the money invested when they thought their investment are capital protected as the financial advisers assured [3]. Moreover, Lehman Brothers did not sell their derivatives directly to individual investors, instead it offered these products to those investors through various local banks of Hong Kong, like Bank of China and Hang Seng Bank, so most of these investors do not even know that the asset they bought is actually guaranteed by Lehman Brothers. Until now, 'Alliance of Lehman Brothers Victims', an organization formed by those investors, is still struggling through various means, including lawsuits and protests, trying to get a refund of capital from those local resale banks [3, 4]. By this tragic case, we learned that ordinary investors do not have any idea on the characteristics of complex products which they are putting money in, even the actual issuer of those products. This comes into questions that, are derivatives products appropriate for individual investors? If not, what kind of investment strategy should be applied as an individual investor?

Through buying a stock and its derivatives with bearish stance, or the opposite way, short selling a stock and buying its derivatives with bullish stance, earnings from those products would be able to cover some of the losses when the movement of that particular stock is out of the investor's expectation. Equity derivatives, therefore, are described as being able to 'hedge' the risk of investing the stock, that we could avoid excessive loss, in exchange, by sacrificing part of the possible return as shown in figure 1.

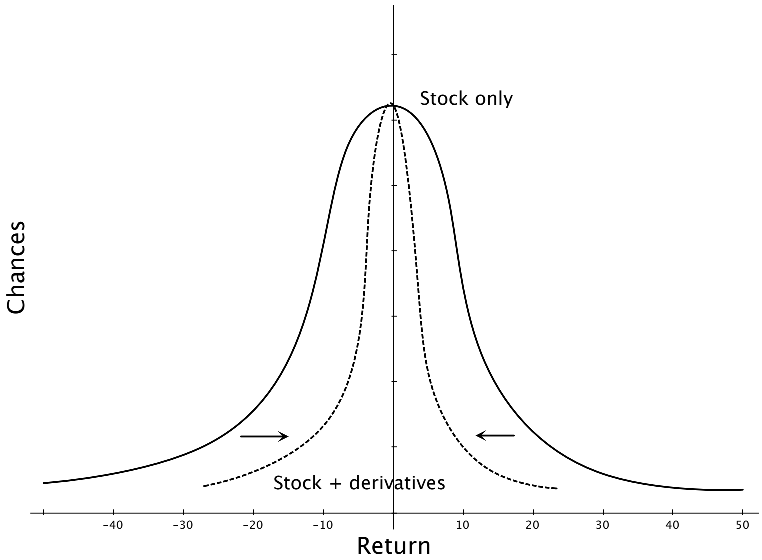


Figure 1: Perceived outcome of hedging

But this is not the end of the story. Although many of these derivatives are traded in exchange like ordinary stocks, they are guaranteed by third-party issuers. It may not be often, but there are still chances that the derivative issuers go bankrupt or break the contacts like the case of Lehman Brothers. Through 'hedging', market risk is reduced, but counter-party risk is created. In fact, one of the strong critics of derivative products are their nature - securitization of the underlying assets, blurs the genuine image of those underlyings from the investors. Also, together with the complexity of derivative terms, like implied volatility[1] and barrier prices[2], making these products not suitable for normal investors.

3.2 Returning to Portfolio Diversification

In the author's opinion, instead of buying derivative products, the technique of portfolio diversification should be used to reduce the risk of buying stocks as an individual investor. There are mainly 2 types of risk in an operating business, they include systematic risk and business risk. Systematic risk, or market risk refers to the factors common to all securities, like economic downturn, and business risk refers to the factors associated with individual assets, like projects that the business is doing, which is diversifiable. As a result, by selecting stocks

[1] Often seen in warrants.

[2] Often seen in CBBCs and other barrier options like accumulators.

with different areas, industries, business cycles and market trends to form a portfolio, price fluctuations of a single stock only contributes a low proportion of it, and thus the return of it will become more stable. Selecting a good portfolio can achieve the same advantage of trading derivatives by reducing all the specific risk of individual firms in the portfolio thus letting only the market risk remains, as shown in figure 2, without suffering from extra counter-party risk or studying complex terms.

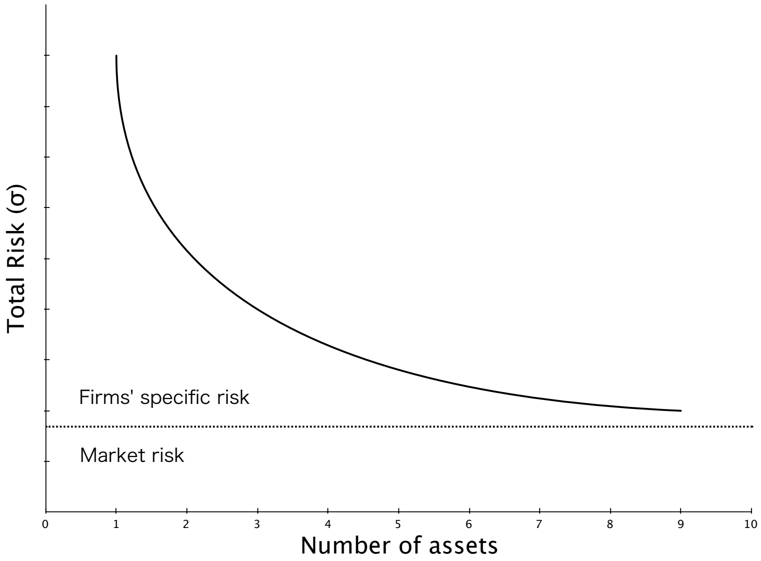


Figure 2: Outcome of portfolio diversification

There are existing theories and tools in mathematical finance for optimizing the proportion of each selected stock in one portfolio in order to achieve the maximum return per unit of risk among all other possible weighting options. However, it is still difficult to decide which stocks should be included in a portfolio. A sophisticated technique which is able to provide this kind of analytics and assist in users' investment decision making, as well as showing the importance of portfolio diversification to the general public, would be very beneficial to both individual investors and the society. By these backgrounds and reasons, the author is proposing this project to research on the application of data mining, to design and develop such a technique and software application.

4 Objectives and Outcome

4.1 Project Objectives

This project is targeted to research on existing data mining algorithms, design and implement a sophisticated software tool which can assist its users when they make investment decision. Users would enjoy the benefit of risk diversification when they study on, and then apply the stock combination advised by the tool.

The study on data mining techniques will also benefit other software developers when they extend the capabilities of this tool or apply the same technique to achieve related goals. Last but not least, as one of the portfolio optimization tools available in the future, it intends to contribute in promoting the practice of diversification and rational investing to the general public.

4.2 Project Scope Description

To accomplish the objectives stated in section 4.1, the following project specifications, scope and boundary are suggested:

- Software specifications
  - Contains an executable command-line component for collection of end-of-day (EOD) stock data. This component will accept input to specify the time frame of data to be collected, and save the data in user desired location.
  - Contains an executable command-line component for collection of current yield of US treasury bills[3]. This component will save the data in user desired location.
  - Contains an executable command-line component which analyzes the stock data collected by data mining. This component will have several data mining algorithm built-in and allow users with relevant knowledge to select their preferred algorithm. The component will store the result in user desired location.
  - Contains an executable command-line component which calculates a set of performance indicators for the stock data collected. The component will store the result in user desired location.
  - Contains a Graphical User Interface (GUI) for user interaction. This component will visualize the analysis result, suggest additional stocks for the user's portfolio and calculate the optimal contribution of each stock in it. Statistical calculation of risk carried by the portfolio, and return per risk, will also be illustrated to user.
- Project scope
  - All stocks listed in main board or growth enterprise market (GEM) in Hong Kong, will be analyzed by the software tool.
  - Price data of these stocks, since their listing, or since the earliest date that the writer could acquire their EOD data, will be analyzed by the software tool.
- Project boundary
  - All assets which are not listed in Hong Kong, unless otherwise stated, are excluded from the scope.

[3] US treasury bills are considered as risk-free asset, their rates will be used in some performance measures of stocks. Refer to section 5.3 for more details.

- All other products listed in Hong Kong, like warrants, ELIs, CBBCs, debt securities, units trusts and mutual funds, are excluded from the scope.
- Real-time and intraday stock prices are excluded from the scope.
- Price data which cannot be acquired through the supported data source stated above, are excluded from the scope.

5 Methodologies

5.1 Collection of Data

The different types of data required by the project are collected from the website listed below:

Data Type	Data Source
Stock Price	Yahoo! Finance
Stock List	HKEx website
Index Price	Yahoo! Finance
US Treasury Yield	Bloomberg website

Table 1: Data source for the project

The above data source is selected because they are either official or very well-established website. They are expected to the stable in terms of availability. However, the writer found that the stock data collected from Yahoo! Finance HK is occasionally incorrect. Historical price data of several stocks consist of price ticks which are observed with price changes when the dates are actually Hong Kong holidays. Since there is no close substitute of Yahoo! Finance in providing the historical prices of Hong Kong's stock, countermeasure of filtering these holidays ticks has been done. They will be discussed in the later sections of this report.

5.2 Data Mining Techniques

As stated in the proposal, the clustering techniques are the key pillars of this project. An effective algorithm is very important for the software users to find out the stocks which are able to provide the greatest diversification on their current portfolio. In this section, the writer is going to discuss the clustering algorithm applied in this project.

5.2.1 Classical K-means Technique

K-means technique is one of the classical clustering algorithms, it focuses on measuring similarity and dissimilarity of data sets. With these similarity figures, grouping the data sets into different clusters would therefore be possible.

Data sets in the same cluster have similar data, where data sets across different clusters will be more distinct.

One of the most common methods for measuring similarity and dissimilarity is euclidean distance. According to Gan, Ma and Wu [5], the formula of calculating euclidean distance is:

$$d_{euc}\left(x, y\right) = \left[\sum_{j=1}^{d} \left(x_j - y_j\right)^2\right]^{\frac{1}{2}}$$

In the formula, x_j and y_j represents the j-th attribute of data set x and y, respectively. The greater the euclidean distance, the higher the dissimilarity between the two data sets.

After we know how the distance between data sets is defined, we need to know how we can group different data sets into clusters. K-means clustering method suggests that we could initially create those clusters by randomly selecting a data set for each of the clusters, and then for the rest of the data sets, we could compare the distance between them and the mean of each clusters, and finally putting the data sets into the cluster where they have the shortest distance.

For its application on the stock data, the software algorithm has followed the below steps:

- Main algorithm

  1. According to the given parameter of number of clusters, randomly select one stock for each of the clusters.
  2. For each stock S randomly selected above, assign its set of return $S = \{R_{s1}, R_{s2,}, \ldots, R_{sT}\}$ to represent the center of the cluster.
  3. For each stock S,
     (a) Calculate the euclidean distance between its set of return and the center of each cluster C, $d_S = \{d(S, C_1), d(S, C_2), \ldots, d(S, C_n)\}$
     (b) Find the minimum element among the set of distance calculated, assign stock S as an element of the cluster C which they have the smallest distance.
  4. For each cluster C,
     (a) Sum up the set of return of stock S inside $C = \{S_{c1}, S_{c2}, \ldots, S_{cn}\}$ for each time t, after that, take an average of them. Formally, $C_{center} = \{\sum_i^n R_{t1}/n, \sum_i^n R_{t2}/n, \ldots, \sum_i^n R_T/n\}$
  5. If there is any changes in the element of clusters, redo the steps starting from step 2.

K-means and euclidean distance are the easiest and most popular way to perform clustering on data sets. However, in the writer's analysis, this algorithm may not be the best way to process time series data. For more details on the evaluation of this algorithm, readers can refer to section5.4 and 8.1.1.

5.2.2 Identical Sequence Extraction

To handle the specialty of sequential data, previous works done by Han and Kamber [6], Ma and Chan [7], suggests that the sequential data can be divided into a number of sub-sequences by sliding a window with a predefined width, across the whole sequence. After that, analysis can be performed on those sub-sequences.

To apply this in historical price analysis, we can define the window by a desired width w and apply this to the price or price change series of stock S:

$$S = \{P_1, P_2, \ldots, P_n\}$$

After that, a set of sub-sequence, with a count of $n - w + 1$, will be produced as follows:

$$S_{sub} = \begin{bmatrix} S_1 = \{P_1, P_2, \ldots, P_w\} \\ S_2 = \{P_{w+1}, P_{w+2}, \ldots, P_{2w}\} \\ \vdots \\ S_{n-w+1} = \{P_{n-w+1}, P_{n-w+2}, P_n\} \end{bmatrix}$$

Once every stocks have their historical price data converted into sub-sequences, we can apply different data mining techniques to discover the interesting patterns among them. This includes clustering and separating different stocks into groups according to the difference of their occurrence on each sequence.

As either the stock price or price change data are in continuous numeric form, a very low matching rate would be observed if the numeric sets are just compared exact numbers. Therefore, certain discretization method on the numeric data are used to divide these continuous data into certain discrete interval, in effect providing an approximated result during the comparison. Readers can refer to section 5.2.5 for more details.

For example, we can discretize the price change number into 3 intervals, they are up, U, down, D, and no change, N. Given this rule, we will be able to produce a discretized price change sequence, like the following:

$$S = \{U, N, N, D, D, D, U, D, N, U, U, N, U, D, D, D, \ldots\}$$

Now we are able to perform the above discussed window sliding steps and count the occurrences of each possible sequence. As a result, letting the window has a width of 5, a matrix like the one below could be formed.

$$O_S = \begin{bmatrix} \{N,N,N,N,N\} & 101 \\ \{N,N,N,N,U\} & 87 \\ \{N,N,N,N,D\} & 81 \\ \vdots & \vdots \\ \{U,U,U,U,N\} & 7 \\ \{U,U,U,U,D\} & 13 \\ \vdots & \vdots \\ \{D,D,D,D,N\} & 11 \\ \{D,D,D,D,D\} & 21 \end{bmatrix}$$

Every possible sequences can act as an attribute in the clustering process. When all stocks have their sub-sequence occurrences counted, we can apply the k-means algorithm on the set of occurrences of each stock. It is because stocks which are considered similar should have similar occurrence count on most of the possible sequence.

The specialty of sequence extraction is the ability to deal with time lag and noises when processing the series that the classical k-means clustering cannot handle. By separating the time series into a set of sub-sequences, even if a stock highly follows another's trend with a small time lag, the clustering algorithm would recognize them as similar stocks and therefore putting them into same cluster.

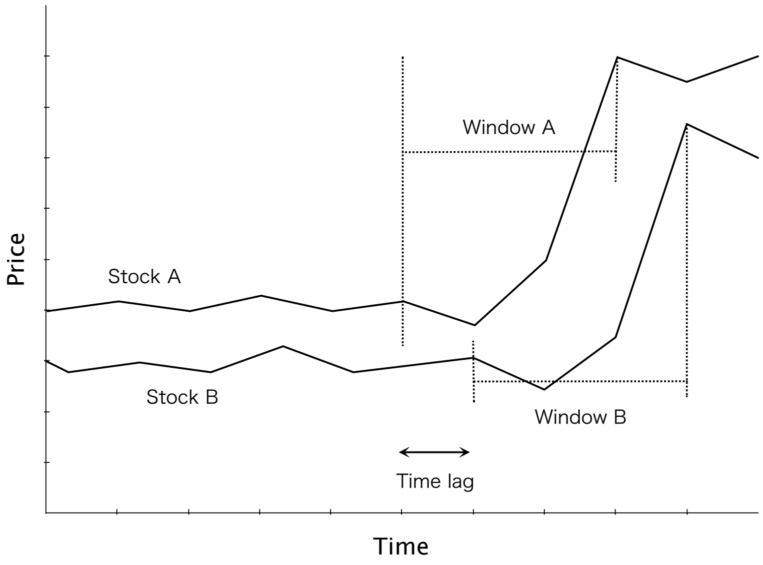


Figure 3: Example of similar series with time lag handled by sequence extraction

In sum, the software deliverable of this project will perform the following steps for this algorithm:

- Data preprocessing

1. Discretize the price data of every stocks.

- Main algorithm
  1. According to the given parameter of width, construct a list with all possible sequence.
  2. For each stock S, slide a window with the defined width, count the occurrences of each possible sequence and form the matrix O_S.
  3. Perform k-means algorithm on the set of O_S acquired.

As the sequential nature of data is handled by introducing the concept of sequence in the algorithm, we can expect that it can produce a more effective clustering result. More details on the evaluation methodology can refer to section5.4. Performance of this algorithm can refer to section 8.1.2.

5.2.3 Similar Motif Discovery

While the previous algorithm divides the sub-sequence of price series evenly by the same width. The principle of motifs are trying to locate exact or approximate matches with longest length possible, among the sets of time series data[8].

Given a 3-interval setting of discretizing the price change series, we will be able to produce some sets of stock sequences, like the following examples of stock a and b:

$$S_a = \{U, N, N, D, U, D, N, U, U, N, U, D\}$$

$$S_b = \{N, N, D, U, D, N, N, U, U, N, U, D\}$$

We can then identify that the longest motif, or the longest consecutive subsequence between S_a and S_b, is $\{N, U, U, N, U, D\}$, which is the tailing subset in the sequence. However, we can also notice a great similarity in the starting part of the sequence, it is not considered just because there is an irrelevant ticks, or noise in between. Therefore, instead of identifying the the longest consecutive sub-sequence, identical sequences which is not linked together should also be considered, as follows:

$$S_a = \{U, \{N, N, D, U, D\}, \{N, U, U, N, U, D\}\}$$

$$S_b = \{\{N, N, D, U, D\}, N, \{N, U, U, N, U, D\}\}$$

The above result shows that the two stocks have a similar motif with time-span equals 11, instead of 6. This value is also known as the length of the longest common sub-sequence.

$$lcs(S_a, S_b) = 11$$

In our application, the algorithm compares each stock with the others using this method, and then build up a similarity matrix to store the length of longest common sub-sequences between different stocks. This matrix will have a size of n^2, where n denotes the number of stocks processed by the algorithm.

$$Similarity = \begin{bmatrix} sim(S_1, S_1) & sim(S_1, S_2) & \cdots & sim(S_1, S_n) \\ sim(S_2, S_1) & sim(S_2, S_2) & & \vdots \\ \vdots & & \ddots & \\ sim(S_n, S_1) & \ldots & & sim(S_n, S_n) \end{bmatrix}$$

The *sim* function denoted above is an extension of *lcs* calculations, it includes a similarity multiplier, m, which is used to magnify the degree of similarities among the stocks, as follows:

$$sim(S_a, S_b) = lcs(S_a, S_b)^m$$

After adding this parameter, a much greater similarity will be observed if two stocks have a very similar or even identical fluctuation patterns. In contrast, a very low similarity will be result from stocks which they have nothing in common at all. The similarity multiplier can be defined by the users.

Like the algorithm of sequence extraction discussed in above section, the motif discovery also designed to be able to handle similar time series with time lag and noises. An illustration is shown below.

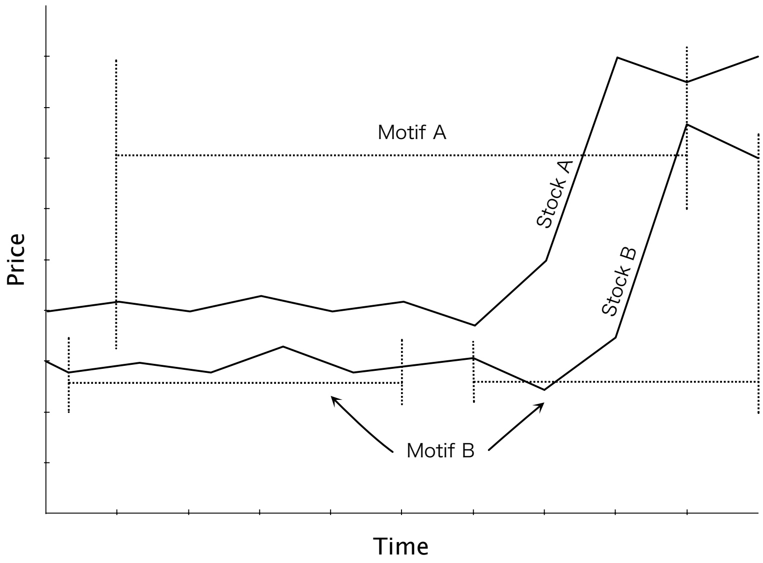


Figure 4: Example of similar series with time lag handled by motif extraction

The algorithm will perform the steps of k-means on the similarity matrix after the matrix has been acquired. This process is meaningful because, if one stock is similar with another stock, their similarity metrics to the rest of the stocks must be similar as well. In sum, the software deliverable will perform the following steps for this algorithm:

- Data preprocessing
    1. Discretize the price data of every stocks.
- Main algorithm
    1. For each stock S,
        (a) Calculate the similarity with all stocks
        (b) Fill the similarity matrix with calculation results
    2. Perform k-means algorithm on the similarity matrix

Further discussion and evaluation on the effectiveness of this algorithm can refer to section 5.4 and 8.1.3 respectively.

5.2.4 Further Clustering on Existing Clusters

As the defined objectives of the software deliverable of this project is to provide a platform for sophisticated clustering analysis on stocks, and also to provide the ease for possible future extension. Further clustering, or any other data mining operations on existing clusters must be supported.

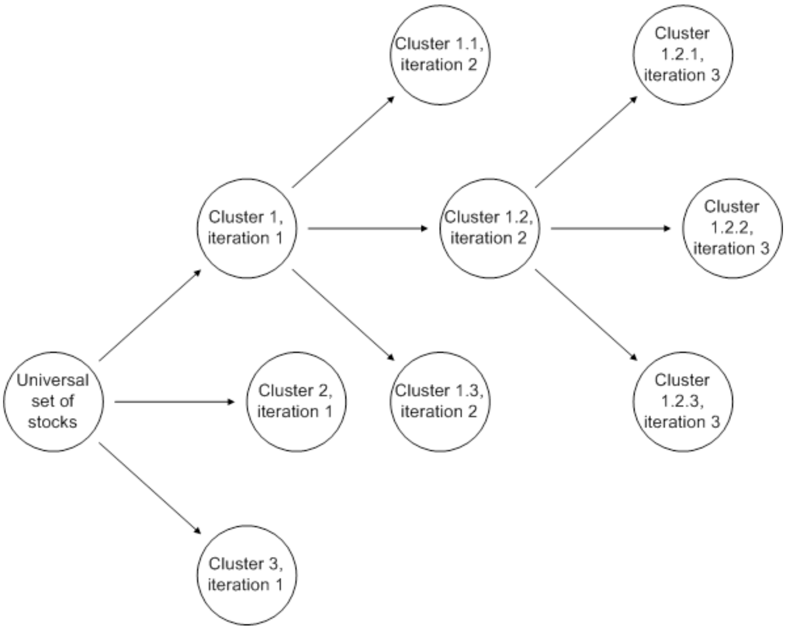


Figure 5: Illustrated process of clustering on existing clusters

5.2.5 Discretization of Price Data

There are currently two discretization method implemented in the software deliverable, they include:

- Discretize by price change
  - By using this option, the continuous numeric data of price change can be discretize into a number of intervals defined by users, to represent different degrees of upside or downside of the ticks of price change.
- Discretize by price level
  - By using this option, the continuous numeric data of price can be discretize into a number of intervals defined by users, in order to represent different levels of price compared with the average price of that particular stock. As this option will become very ineffective in processing prices with a very long time span due to the long term growth rate of the stock, another parameter named as referencing period is provided for users. Users can define the length of this period, after that, the price ticks will be compared with the average of their corresponding referencing period, instead of comparing with the all-time average.

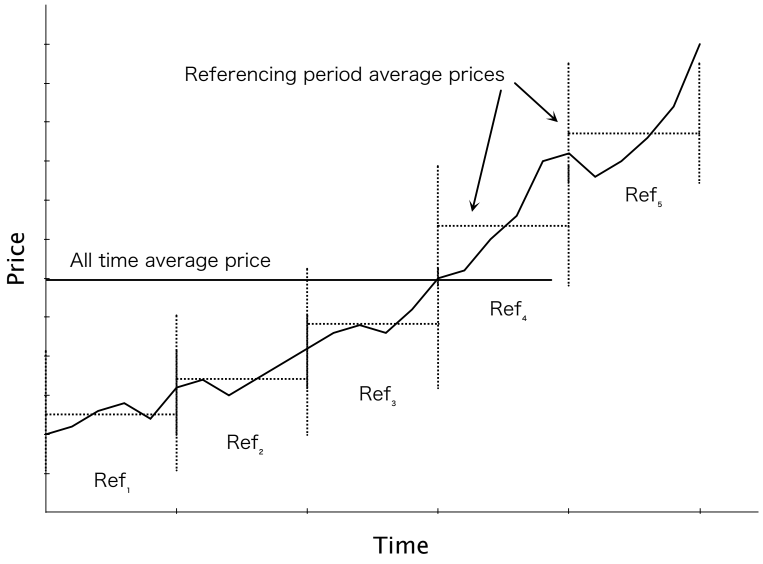


Figure 6: Illustrated average price calculation by separating the whole period into referencing periods

5.3 Portfolio Optimization

After some desirable stocks have been shortlisted, there are certain metrics for us to review and then select the best one among the stocks. Also, there are methods for us optimize the proportion of each stock in the portfolio, so that we can achieve a weighting which has the lowest risk but the highest return among the possible options. In this section, the writer is going to discuss the related techniques studied so far.

5.3.1 Modern Portfolio Theory

Developed by Markowitz [9, 10], Modern Portfolio Theory (MPT) is the first and also the most important foundation of mathematical techniques in portfolio optimization. It introduces the analysis of investment portfolios by considering the expected return and underlying risk of each individual assets and, crucially, the interrelationship of these assets in the portfolio. It provides a mathematical framework for quantifying risk and return, enables comparison between portfolios using these quantitative measurements. Before this, investors can only examine their investments one by one through fundamental or technical analysis, and then build up portfolios of their favored stocks without the concern of their relationship in between. Markowitz's contribution is a breakthrough in both the mathematical and finance areas at that time.

In general, risk and return are positively proportional. This means investments which have higher risk are expected to have a higher return. For example, investing on the real estate market would have a higher return than stock market, as the relatively lower market liquidity creates extra uncertainty, or risk, which requires a higher return to compensate. As a result, if there are two portfolios that offer the same expected return, there is no reason for investors to select the more risky one. With the belief on this condition that investors are risk averse, MPT calculates and compares the return per risk of given portfolios, so that investors can review these statistics and select a better portfolio.

The theory considers a portfolio as a weighted combination of its assets, and thus the return of a portfolio is the weighted return of its assets. Therefore, the expected return of portfolio can be expressed as the following formula:

$$E(R_p) = \sum_i w_i E(R_i)$$

R_p represents the return of the portfolio, R_i represents the return of individual stock, or investment i, w_i represents the the proportion of asset i contributed to the portfolio.

The theory also models the return of an investment as normally distributed and the risk of it as the standard deviation of return, and the variance of portfolio return is be expressed as:

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij}$$

The second part of the formula is considered as a model of the interrelationship of two stocks and how this relationship affects the return of the portfolio, where ρ_{ij} is the correlation efficient between the return of the two stocks, i and j. And the standard deviation, or the risk of the portfolio, can be simply calculated by taking a square root of the variance:

$$\sigma_p = \sqrt{\sigma_p^2}$$

If an investor is provided with various portfolios for him to choose, he or she can apply the formula above to calculate the return and risk of each portfolio and plot the following graph:

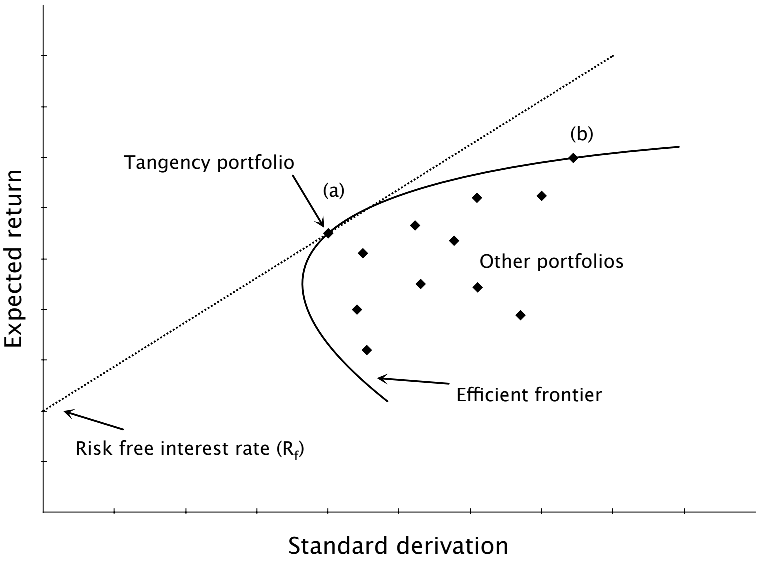


Figure 7: Efficient frontier and tangency portfolio

The curve which intercept with the straight line started from risk free interest rate, is called the efficient frontier. The portfolio represent by point (a) can offer the greatest return per risk with a specific asset selection and weighting applied on each individual asset. That particular portfolio and weighting of stocks is the best option available for the investor. In contrast, the portfolio represent by point (b) has the same stock selection as (a), but a different weighting contribution on its assets. Although portfolio (b) could achieve higher expected return than (a), it has much higher risk, resulting a lower figure of return per risk compared with (a). MPT further suggests the separation theorem, that all investors will go for portfolio (a) indifferently, if they are rational. The difference of risk aversity of investors would only make them to select different portfolios lies on the tangency line, which represents portfolios combined by the tangency portfolio (a) and the risk free asset with different weightings. The investors can simply reduce the holdings of this optimal portfolio and purchase risk free assets if lower risk is desired, or simply borrow risk free assets to increase the holdings of this portfolio if higher risk and return is desired.

In the above illustration, we assumes the investors are able to borrow at risk free rate. In reality, it is nearly impossible to be the case. However, even if we loosen this assumption of classical MPT. The tangency portfolio remains to be one of the good portfolios within all other possibilities.

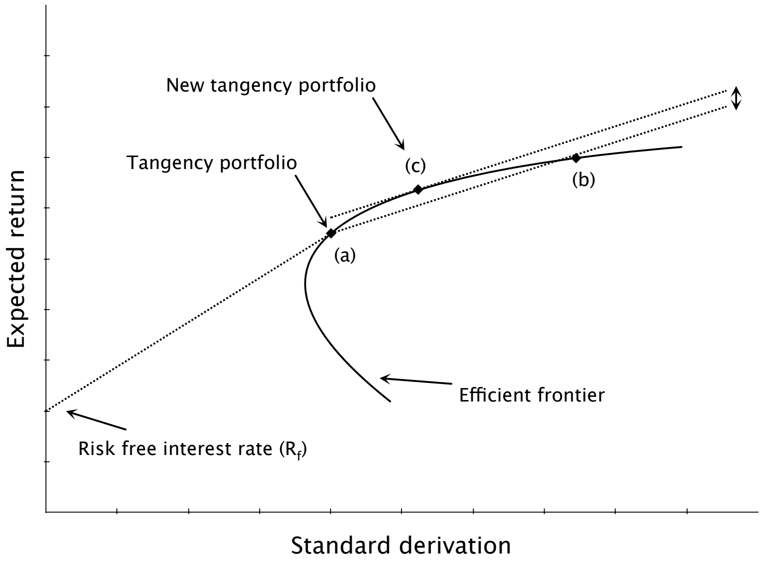


Figure 8: Efficient frontier and tangency portfolio with increased borrowing rate

As we can see in the above figure, the higher part of the tangency line has a relatively gentle slope compared with the lower part of the line. Under this case, the return per risk ratio of portfolio (a) and (b) are the same, as investors borrow more asset and purchase portfolio (b), they recreates the risk and return level of portfolio (b). Although a new tangency portfolio represented by point (c) can be found, the differences between the portfolios are limited. Therefore, if the interest rate spread of lending and borrowing remains small, the return per risk analysis has strong indication for our decisions in portfolio selection.

Instead of plotting out figure 7, the optimal portfolio can also be found by finding the maximum of return per risk of all available portfolios, the return per risk of a certain portfolio can also be calculated by the following formula:

$$S = \frac{E(R) - R_f}{\sigma}$$

This is also called the Sharpe ratio. $E(R_f)$ indicates the expected return offered by the risk free asset.

In order to find the portfolio weighting which provides the highest Sharpe ratio, we may take a derivative of the formulae discussed, resulting the below optimal portfolio equation for a 2 assets portfolio formed by asset i and j:

$$w_i = \frac{[E(R_i) - R_f]\,\sigma_j^2 - [E(R_j) - R_f]\,\sigma_i\sigma_j\rho_{ij}}{[E(R_i) - R_f]\,\sigma_j^2 + [E(R_j) - R_f]\,\sigma_i^2 - [E(R_i) - R_f + E(R_j) - R_f]\,\sigma_i\sigma_j\rho_{ij}}$$

While the optimal portfolio equation of a 2 assets portfolio remains solvable, finding the optimal weightings of asset in a portfolio with many assets cannot be done by a simple equation.

MPT is considered effective to select an optimal portfolio among a few possible options, or deciding an optimal proportion for the given stocks inside the portfolio. However, it becomes extremely difficult and time consuming, even though it is possible, to decide whether a stock should be included in a portfolio to achieve the maximum Sharpe ratio, because of its nature of requiring a massive number of factor inputs for calculation. Like Markowitz described [10], this theory is focus on the choice of portfolio, instead of the observation on the stocks.

In the system built by this project, the application of modern portfolio theory is limited in the optimization of assets' weightings in a portfolio. For 2 assets portfolio, the above mentioned formula will be used. For portfolio having more than 2 assets, the method of trial and error are used.

5.3.2 Capital Asset Pricing Model

The Capital Asset Pricing Model (CAPM), introduces another risk metrics, beta (β), which is also commonly used nowadays [11] in addition to the standard deviation of return. The beta measures the responsiveness of a stock to movements of the market portfolio, which is considered as a combination of all kinds of possible financial asset, and therefore effectively diversified and consist of only the systematic risk of the market.

Beta is a representation of both the risk and return of an asset, so if a stock is considered to have a high beta, not only its risk is high compared with the market portfolio, its expected return, according to the model, is high too. An asset's beta can be obtained by performing linear regression on the historical return of that particular asset versus market portfolio.

For instance, figure 11 shows the regression chart of China Unicom versus the Hang Seng Index (HSI). The reason for choosing HSI is that the market portfolio could have very board coverage, making us difficult to estimate its performance. Therefore, for the tool developed and also this report, Hang Seng Index (HSI) is selected as a proxy of the market portfolio as the project is mainly concerned about the stocks in Hong Kong. In this example, the beta of China Unicom is found to be 1.4017.

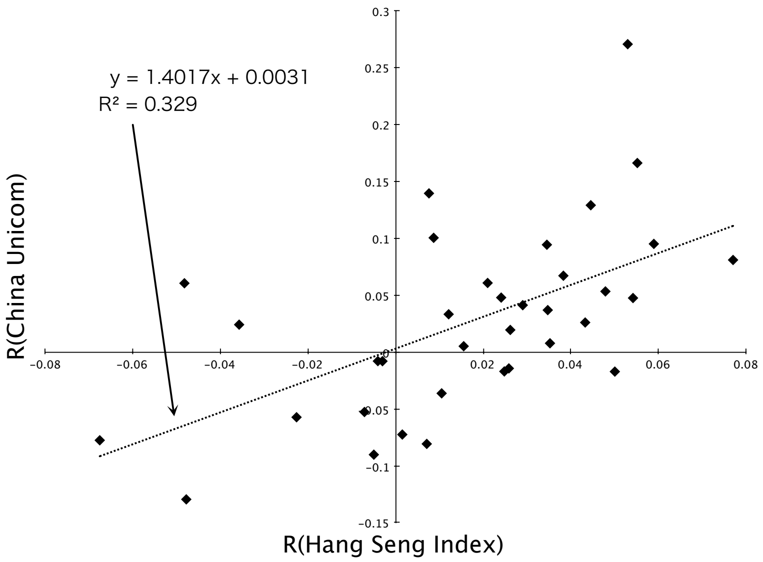


Figure 9: Linear regression on China Unicom's return against HSI

Alternatively, the asset's beta can be calculated by dividing the covariance between asset and market return, by the variance of market return:

$$\beta = \frac{Cov(R, R_M)}{\sigma^2_{R_M}}$$

In this formula, R represents the return of the asset and R_M represents the return of the market portfolio.

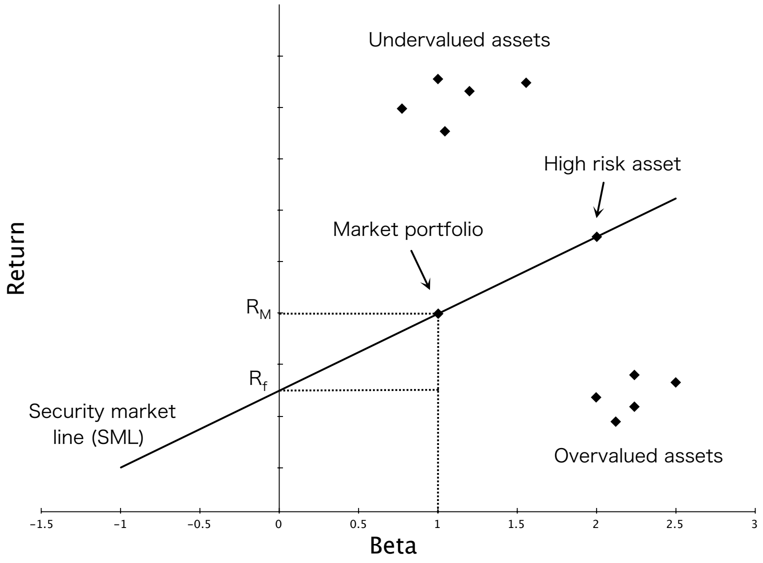


Figure 10: Illustrated graph of CAPM theory and Security Market Line (SML)

According to the CAPM model, the line which connects the point of risk free asset and market portfolio is the Security Market Line (SML). The beta of market portfolio is defined as 1. Return of All other assets are expected to move along SML according to their own beta. These are the fair returns of the assets provided by CAPM, which can be computed by the following equation:

$$E(R) = \beta\left(R_M - R_f\right) + R_f$$

The stocks which has their return equal to the expected return, or put it another way, laying on SML, are correctly priced at the moment. However, for the stocks which are not laying on SML, they are either undervalued or overvalued. These abnormal return are measured by alpha (α) in the following formula:

$$R = E(R) + \alpha$$

Stocks having a positive alpha indicate that they have outperformed the market or undervalued according to CAPM. It is because that particular stock is expected to be bidded up by rational investors, making the stock price back to the normal level which generates only fair return. In contrast, stocks having a negative alpha mean they have underperformed or overpriced. Their price are expected to go down. Stocks which are currently undervalued are quite likely worth investing because their expected return higher than others which have the same amount of risk. In figure 11's example, the alpha of China Unicom is 0.0013.

Beta has many similarities compared with the measures of standard deviation. However, they serve slightly different purposes. While standard deviation measures total risk of an asset, beta represents the sensitivity of the asset to the market movement, thus only measures the market risk of the asset. In figure 2 of section 3.2, we know that the specific risk of individual firm will be reduced as the portfolio become more and more diversified. Therefore, beta and alpha would be particularly useful if we are trying to measure risk and return of portfolios, or individual assets given that a well-diversified portfolio is already in hand. Portfolio beta is identical to the weighted average of betas of its assets:

$$\beta_p = \sum_i w_i \beta_i$$

To illustrate the difference between total risk and market risk, another regression of China Mobile, a state-owned telecommunications firm as same as China Unicom, against HSI has been done. The regression results are shown in the following chart.

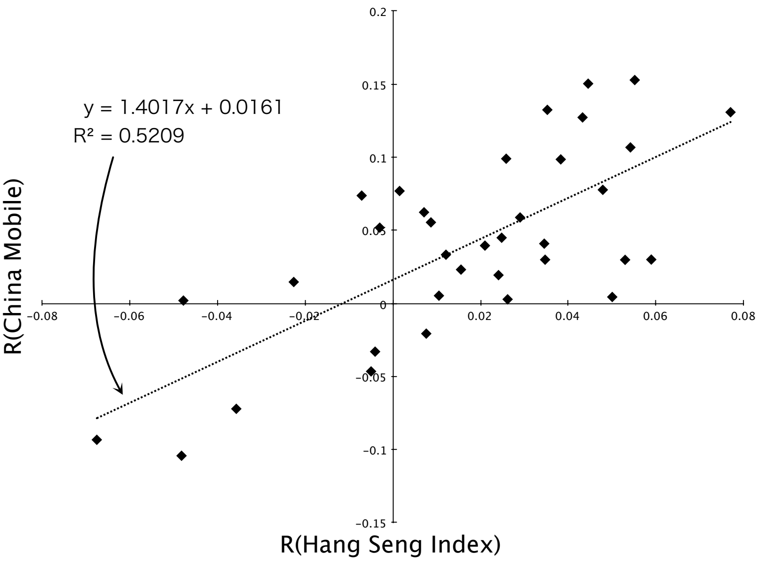


Figure 11: Linear regression on China Mobile's return against HSI

Referring to both figure 9 and 11, we can observe that both stocks have the same beta, 1.4017. However, if we look into the distribution of their periodic returns, we can see that the distribution of China Unicom is more deviate from the characteristic line, thus a higher standard derivation, or total risk, of China Unicom is expected.

By providing these metrics, users of MineStock Workbench can evaluate the stocks found by the data mining algorithms very easily. They are, therefore, allowed to compare and select their own preferred stocks for portfolio diversification.

5.3.3 Value at Risk

Value at Risk (VaR) is a popularly used method to measure the market risk of the financial assets. It is considered as a threshold value such that the mark-to-market (MTM) loss of one asset over the given time period, in a given probability level, exceeds this value.

For example, if a portfolio has a one day 5% VaR of $1,000, there is a 5% probability that the specific stock will fall by value by more than $1,000 over a one day period, assuming the market is in normal condition and there is no trading on the portfolio within the day.

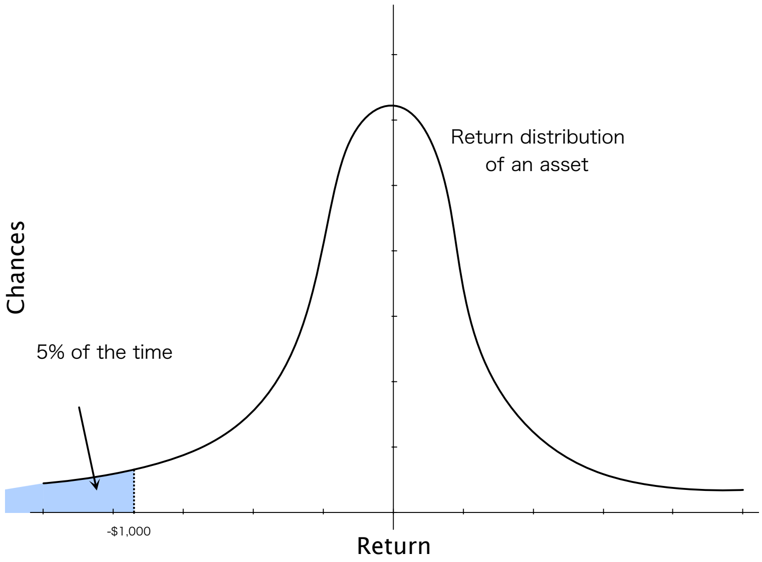


Figure 12: Illustrated concept of Value at Risk

According to Choudhry [12], there are three ways to evaluate the VaR of an asset, as follows:

- Delta normal method
  - Also known as the parametric method. It assumes the distribution of return is normal, and uses standard deviation as a parameter to calculate VaR.
- Historical method
  - It uses the actual return distribution from the historical data, and measure the amount of loss with the given percentage of occurrence.
- Monte Carlo simulation
  - It uses a random walk function to simulate stock price, and then measure the amount of loss with the given percentage of occurrence, using the simulated set of data.

Although all the three methods are widely used in the financial industry, in a recent study performed by the the writer and Siu, et al [13], indicates that their calculation results on one asset could be very different in some exceptional cases, especially when the stock has a long term upward trend in the referencing period. In these cases, even the standard deviation of the stock return is high, resulting in the parametric VaR is high, it overstate the possible amount of loss compared with the historical method, given the same time interval and confidence level.

In this project, after the clustering of stocks has been done, the users can review their current portfolio and consider adding stocks from other clusters in order to enjoy the effect of diversification. The software deliverable of this project will provide both the delta normal VaR and historical VaR of each stock for user reference, such that a stock with less occurrence of downside or big loss could be easier to locate.

5.4 Evaluation of Results

As this project is aimed to assist investors to diversify their stock portfolio and reduce the risk that they are exposing to. The goal of our evaluation process is to ensure the clustering result delivers could achieve these objectives. In section 5.3.1 we discussed MPT, which proposes the usage of standard deviation from return as the measurement of total risk and compiled a formula to calculate the total risk of portfolio. The formula requires the weighting and standard deviation of each individual assets in portfolio, and most importantly, the correlation (ρ) between those different assets.

The degree of correlation of stocks are crucial for reducing specific risk of firms, while a low or negative correlation produces a great effect of diversification, a high correlation of stocks could eliminate no risk at all, as illustrated in the following figure of a 2 stocks portfolio example.

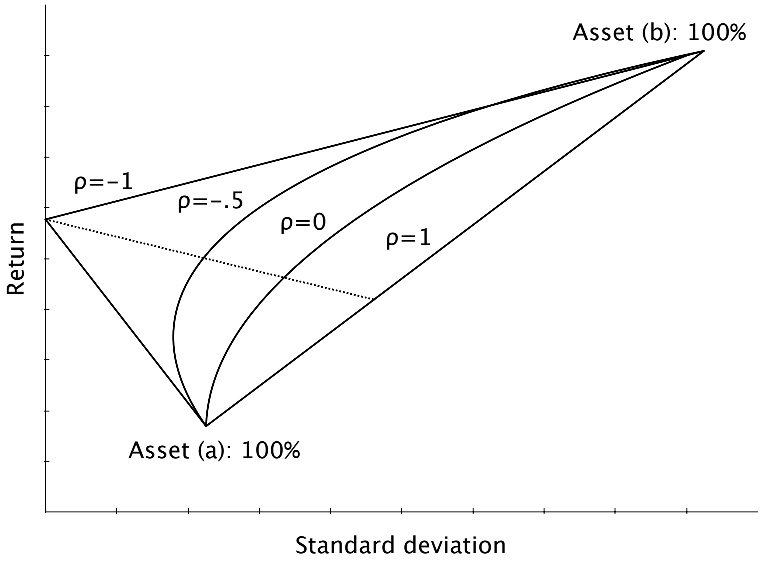


Figure 13: Relation of stocks correlation and diversification effect

As the lowest possible correlation between 2 stocks, -1, implies that their movement is in exactly opposite. Therefore, under certain weighting, their downside fluctuations would be completely offset with a zero-risk return remains. However, a correlation of 1 implies that their movement follow each other. In this case, no diversification effect is possible.

In MineStock Workbench, after the clustering process are done, users will pick the better performed stocks in different clusters to build up their portfolio. By this reason, a higher correlation of stocks in one cluster and a lower correlation of stocks across clusters are needed. In our evaluation section of report, we will compute the intra-cluster correlation and inter-clusters correlation of stocks to verify whether the clustering algorithm is functioning as expected. We will also compare the figures of the 3 clustering algorithm mentioned above to rank their performance.

The intra-cluster correlation and inter-clusters correlation of stocks and clusters will be calculated following the below steps:

- Market correlation
    1. Calculate and sum up the correlation of all the possible stocks pair, $\sum cov(R_{ai}, R_{aj}), \forall a_i \in U, \forall a_j \in U, i \leq j$
    2. Count the number of stocks pair
    3. Divide the sum by the count of stocks pair, get the average correlation of the market
- Intra-cluster correlation
    1. For each cluster C located
        (a) Calculate and sum up the correlation of all the possible stocks pair, $\sum cov(R_{ai}, R_{aj}), \forall a_i \in C, \forall a_j \in C, i \leq j$
        (b) Count the number of stocks pair
        (c) Divide the sum by the count of stocks pair, get the average correlation within the cluster
        (d) Count the number of stocks assigned in that particular cluster
    2. Calculate the average of correlation within clusters, weighted by the number of stocks in each cluster
    3. If any further clustering on the larger clusters has been done, repeat steps 1 and 2 on those sub-clusters.
- Inter-cluster correlation
    1. For each cluster C_a of all clusters including sub-clusters
        (a) For each of all other clusters C_b located including sub-clusters, $\{C_a, C_b\}, a \neq b$
            i. Calculate and sum up the correlation of all the possible stocks pair, $\sum cov(R_{ai}, R_{bj}), \forall a \in C_a, \forall b \in C_b$
            ii. Count the number of stocks pair computed for this cluster pair
            iii. Divide the sum by the count of stocks pair, get the average correlation of the 2 cluster pair
        (b) Calculate the average of correlation of C_a with other clusters, weighted by the number of stocks in each of the other clusters
        (c) Count the number of stocks in each cluster C_a
    2. Calculate the average of inter-cluster correlation, weighted by the number of stocks in each cluster

6 System Architecture and Design

6.1 Development Platform

The system is written by C# and based on Microsoft .NET Framework 4.0 in order to enjoy the benefits of simple and intuitive GUI development. Also, it relatively faster speed compared with Java enables the system to perform the complicated data mining algorithm more effectively.

6.2 Modulized System

To ensure the extensibility of the system, all of its key functions are built as standalone command-line programs. The GUI is programmed to call those command-line programs without directly mutating the data store. Instead, the GUI only access the data store for display purpose. Therefore, the business logic modules, for instance, clustering module of the system can be easily swapped or modified without a great impact to the whole system. The modules can also be easily plugged into other systems as well. Figure 14 illustrates the system's data flow in the context of Model-View-Controller (MVC).

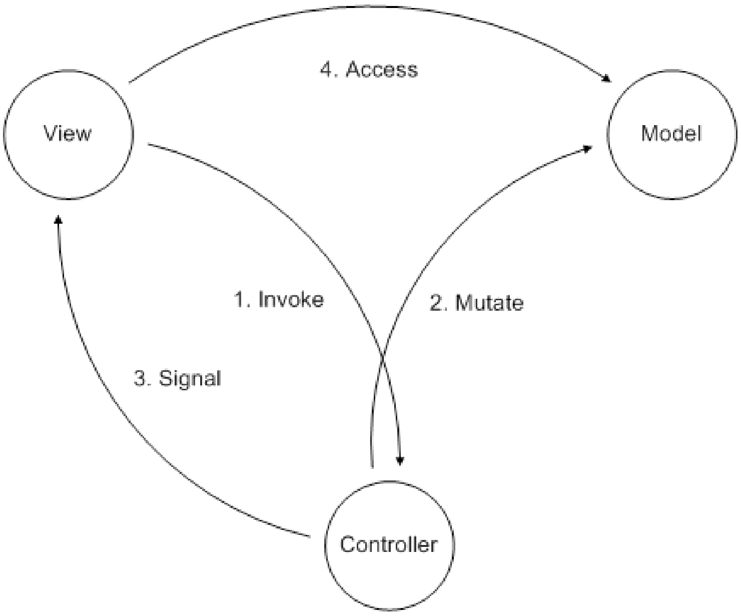


Figure 14: Separation of Model-View-Controller of MineStock Workbench

The command-line components of the system and their data flow are listed in figure 15. While the users only send their commands to the GUI, the GUI act as a dispatcher of commands, executing the command-line modules to manipulate the data store. Detailed information of each command-line modules' data flow are described below:

1. Get stock data module

    - Accept input of desired time period
    - Collect historical price data of stocks from the internet
    - Store the stock data into the data store of stock price

2. Get index data module

    - Collect historical price data of index from the internet
    - Store the index data into the data store of index price

3. Get treasury bill data module

    - Collect current yield of treasury bill from the internet
    - Store the treasury bill data into the data store of stock treasury yield

4. Filter holiday ticks of stock module

    - Accept input of a list of stocks from the data store of stock price, plus one referencing stock or index.
    - Remove data ticks of the stocks in the list if the date of data ticks not exist in the referencing stock or index.
    - Store the updated stock data into the data store of stock price

5. Change stock price time interval module

    - Accept input of a list of stocks from the data store of stock price, plus the input of time interval that is desired to change into.
    - Combine the price ticks of stocks according to the specified interval
    - Store the updated stock data into the data store of stock price

6. Discretize stock data module

    - Accept input of a list of stocks from the data store of stock price, plus the input of the desired algorithm to use.
    - Perform discretization on the stock price with the specified algorithm
    - Store the discretized stock data into the data store of discretized stock price

7. Clustering module

    - Accept input of a list of stocks from the data store of stock price or discretized stock price, plus the input of the desired algorithm to use.
    - Perform clustering on the on the stock price with the specified algorithm
    - Store the result into the data store of clusters

8. Calculate performance indicators module

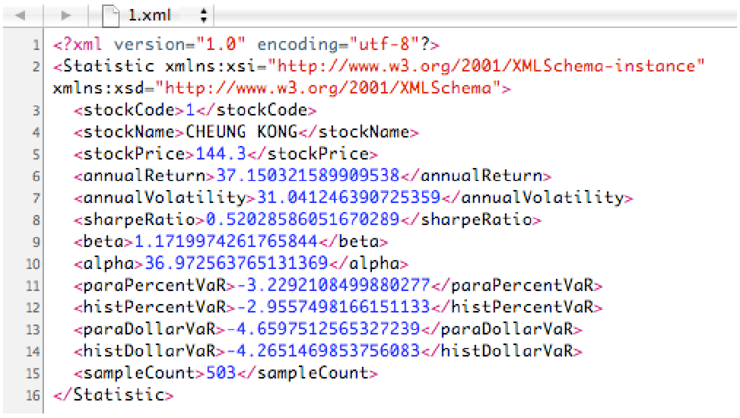
- Accept input of a list of stocks from the data store of stock price, index data from the data store of index price and treasury bill data from the data store of treasury price
- Calculate the statistics of stocks and performance indicators
- Store the result into the data store of statistics

9. Portfolio optimization module

    - Accept input of stock statistics from data store of statistics, portfolio information from portfolio data store
    - Calculate the optimized portfolio weighting
    - Store the result into the data store of portfolio

6.3 XML Data Storage

In order to enjoy the benefit of lightweight data storage and take advantage of the serialization facility provided by the .NET Framework, all data store of the system is in XML format located in the users' own PC. The following figure shows an example of XML file, storing the statistics and performance indicators of Cheung Kong.



```xml
<?xml version="1.0" encoding="utf-8"?>
<Statistic xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <stockCode>1</stockCode>
  <stockName>CHEUNG KONG</stockName>
  <stockPrice>144.3</stockPrice>
  <annualReturn>37.150321589909538</annualReturn>
  <annualVolatility>31.041246390725359</annualVolatility>
  <sharpeRatio>0.52028586051670289</sharpeRatio>
  <beta>1.1719974261765844</beta>
  <alpha>36.972563765131369</alpha>
  <paraPercentVaR>-3.2292108499880277</paraPercentVaR>
  <histPercentVaR>-2.9557498166151133</histPercentVaR>
  <paraDollarVaR>-4.6597512565327239</paraDollarVaR>
  <histDollarVaR>-4.2651469853756083</histDollarVaR>
  <sampleCount>503</sampleCount>
</Statistic>
```

Figure 16: Sample XML file storing the performance indicators of Cheung Kong

7 User Interface

7.1 Layout and Menus

The layout of MineStock Workbench is shown in figure 17. It mostly follow the traditional layout of Microsoft Windows Series and other applications based on Windows. Therefore, users are able to accommodate the interface of this system easily. The functions provided by the system can be divided by 4 main categories listed in figure 18, including downloading and preprocessing data,

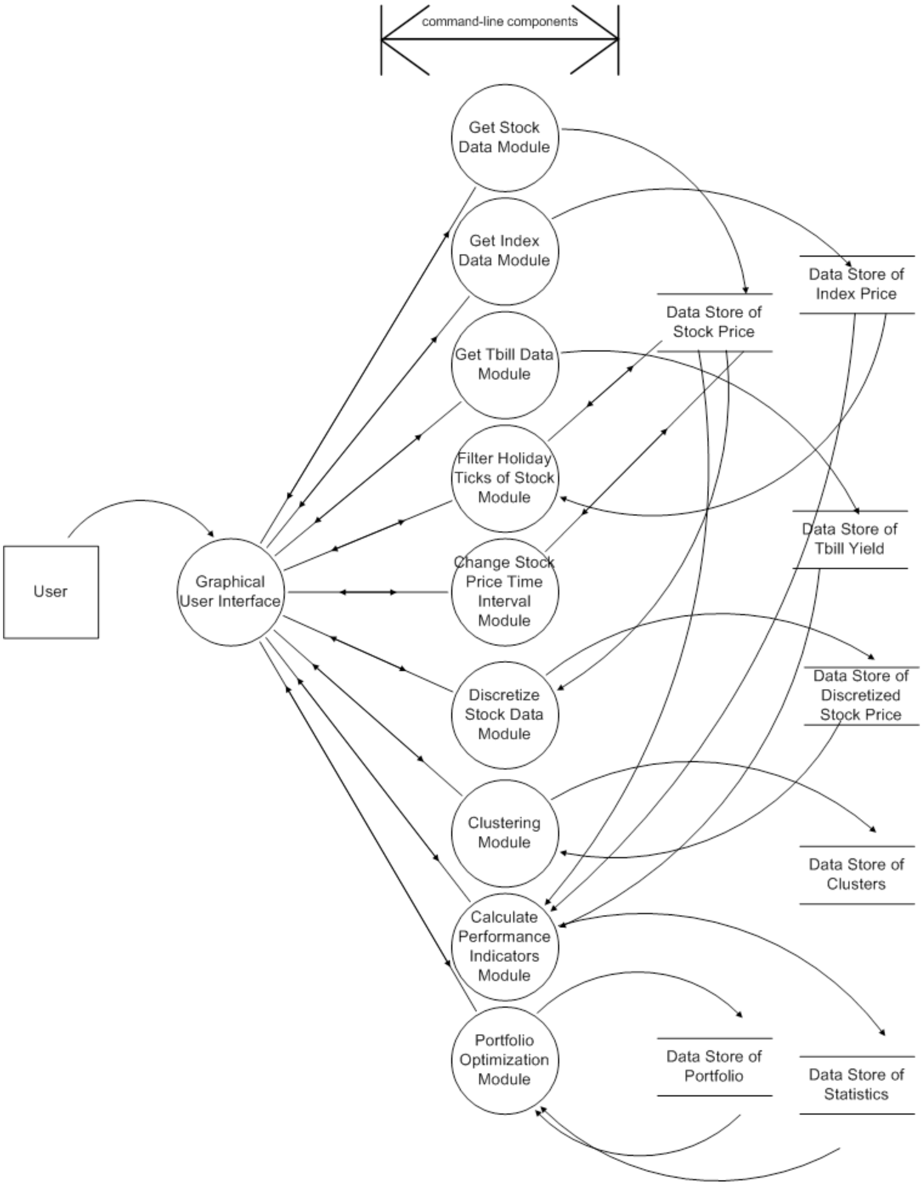

command-line components
Get Stock Data Module
Get Index Data Module
Get Tbill Data Module
Filter Holiday Ticks of Stock Module
Change Stock Price Time Interval Module
Discretize Stock Data Module
Clustering Module
Calculate Performance Indicators Module
Portfolio Optimization Module
User
Graphical User Interface
Data Store of Index Price
Data Store of Stock Price
Data Store of Tbill Yield
Data Store of Discretized Stock Price
Data Store of Clusters
Data Store of Portfolio
Data Store of Statistics

Figure 15: Data Flow Diagram between modules of MineStock Workbench

monitoring functions, analyzing functions, and also the configuring functions. In the following sections, the writer will discuss these 4 main categories of functions and the way to use them in the system.

7.2 Downloading and Preprocessing Functions

The system requires 3 types of data to operate, they are historical price data of stocks, index, and also the current yield of treasury bills.

7.2.1 Stocks and Index Price

To download stock price or index price in the system, users can select 'Data', 'Update Data', then 'Stock...' or 'Index...' to open the panel for downloading data. Their panel are very similar as shown in figure 19. In the panels, inputs including the system paths for storage of stock, index and log files, starting date and ending date of historical prices are required. For downloading stocks, stock code or types of stocks are required to be given or selected. After clicking the 'Start' button, the system will start to fetch data from Yahoo! Finance HK.

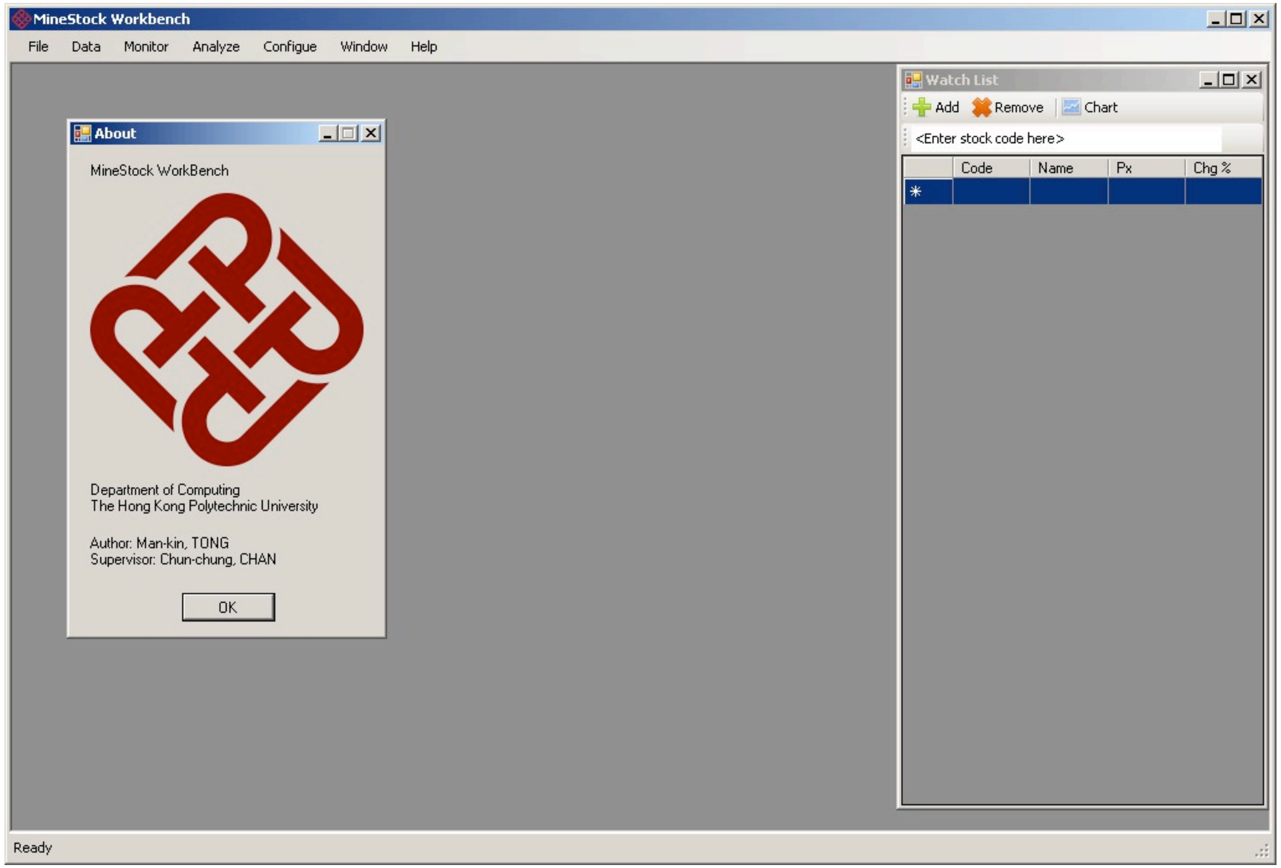
7.2.2 Treasury data

To download treasury data in the system, users can select 'Data', 'Update Data', then 'Treasury...' to open the panel. In the panel, which is identical to the one shown in figure 20, the system paths of data storage and logging are required. After clicking the 'Start' button, the system will start to fetch data from Bloomberg.

7.2.3 Holiday Ticks and Time Interval of Data

The stocks price downloaded by the panel shown in figure 19 is EOD data and sometimes may happen to have price information in non-trading day. To eliminate these holiday price data and ensure all the stocks have the same set of price dates for comparison, users can select 'Data', 'Preprocess Data', then 'Filter Holiday Ticks...'. A panel same as the left one in figure 21 will be shown. In the panel, inputs including the system paths for storage of stock, log files, and also one referencing stock or index are needed. The referencing stock or index will be considered to have the correct price dates, all other stocks, therefore, will be adjusted accordingly after the users clicked the 'Start' button.

To change the time interval of stock and index price, for example, from EOD to month-end, users can select 'Data', 'Preprocess Data', then 'Change the Data Period...'. A panel same as the right one in figure 21 will be shown. In the panel, input and output path of stocks or index data files, also the system paths for log files, and most importantly, the time interval to be converted into, are needed to be entered. The system will then read all the stocks and index files located in input directory, covert them into new time interval and finally store the new data in the output directory after the users clicked the 'Start' button.

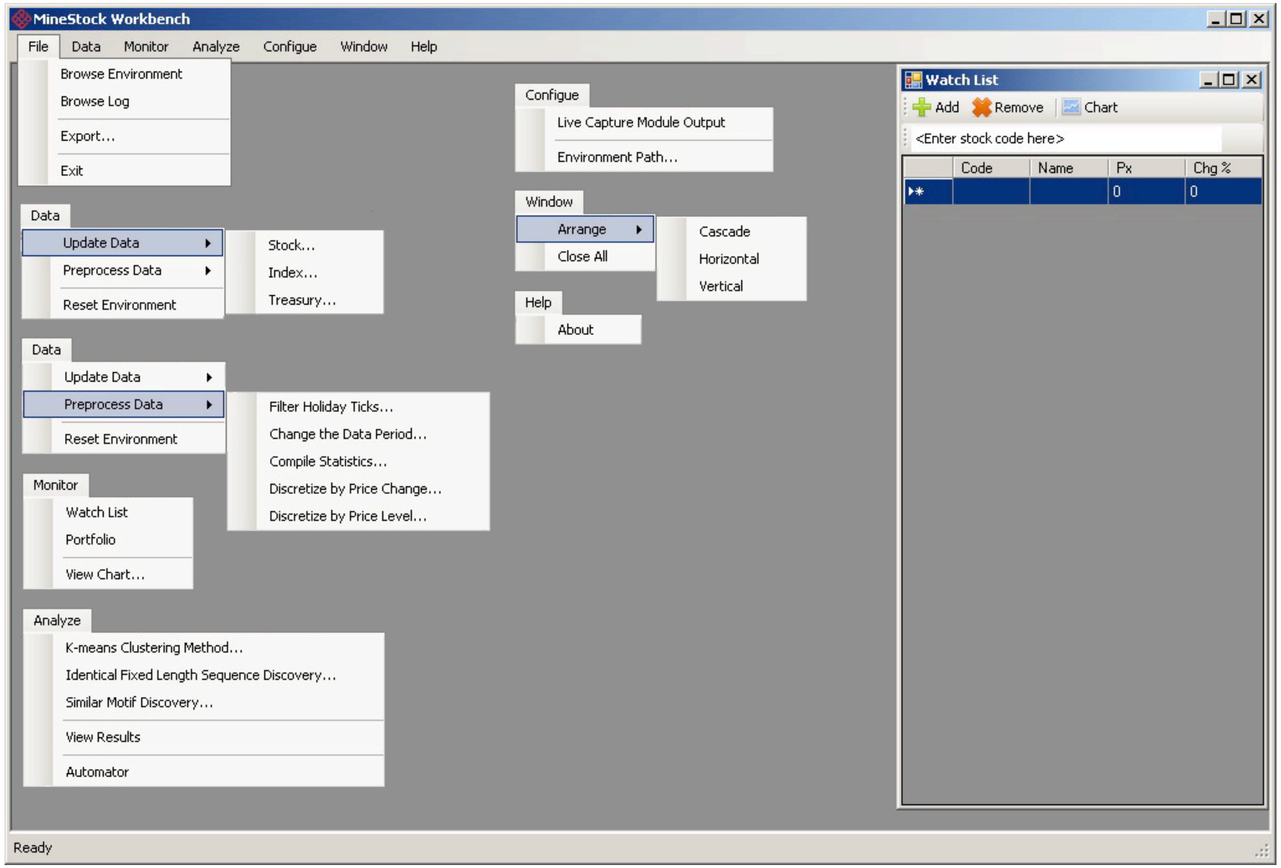


MineStock WorkBench

Department of Computing
The Hong Kong Polytechnic University

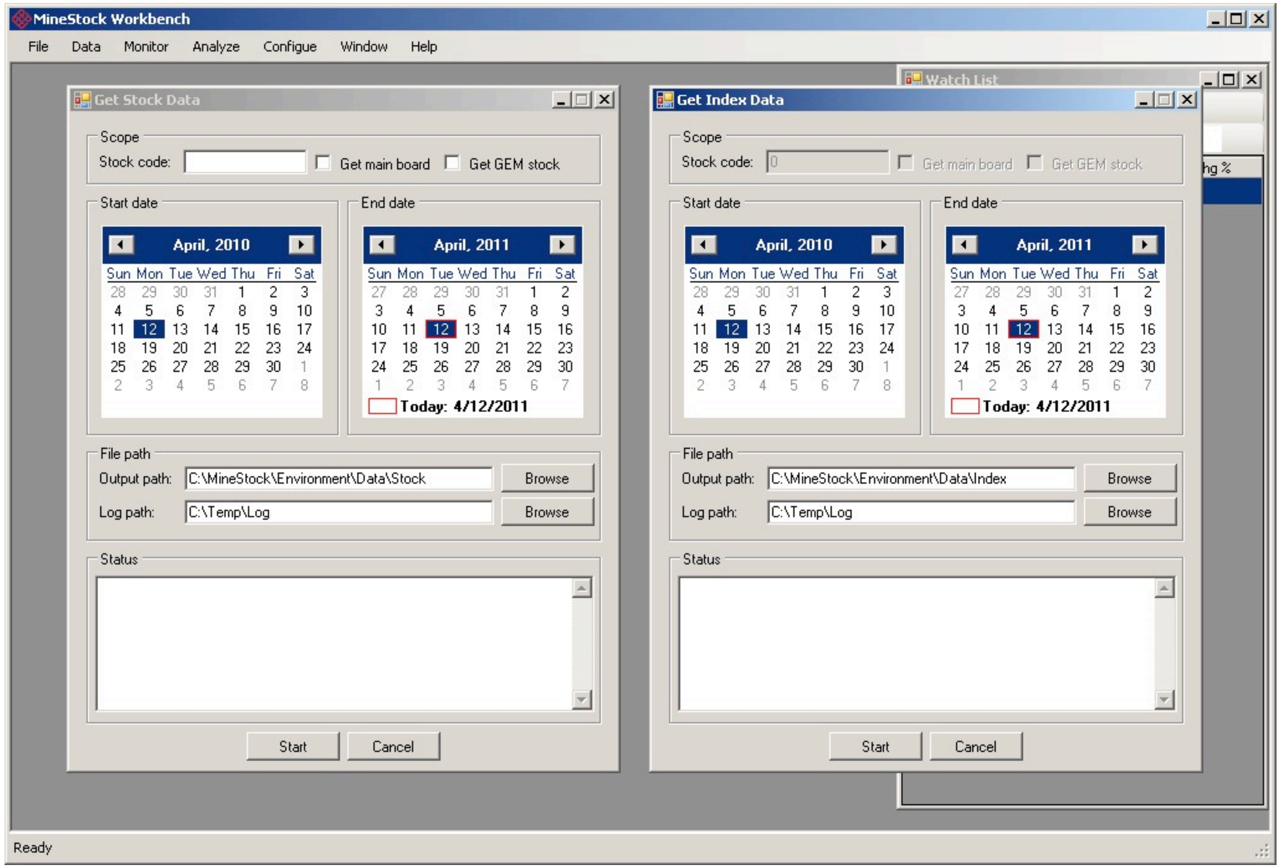
Author: Man-kin, TONG
Supervisor: Chun-chung, CHAN

Figure 17: General layout of MineStock Workbench



	Code	Name	Px	Chg %
▸*			0	0

Figure 18: Functions available in MineStock Workbench



Get Stock Data

Scope

Stock code:

☐ Get main board ☐ Get GEM stock

Start date

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

April, 2010

End date

April, 2011

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Today: 4/12/2011

File path

Output path: C:\MineStock\Environment\Data\Stock

Log path: C:\Temp\Log

Status

Get Index Data

Scope

Stock code: 0

☐ Get main board ☐ Get GEM stock

Start date

April, 2010

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1
2	3	4	5	6	7	8

End date

April, 2011

Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Today: 4/12/2011

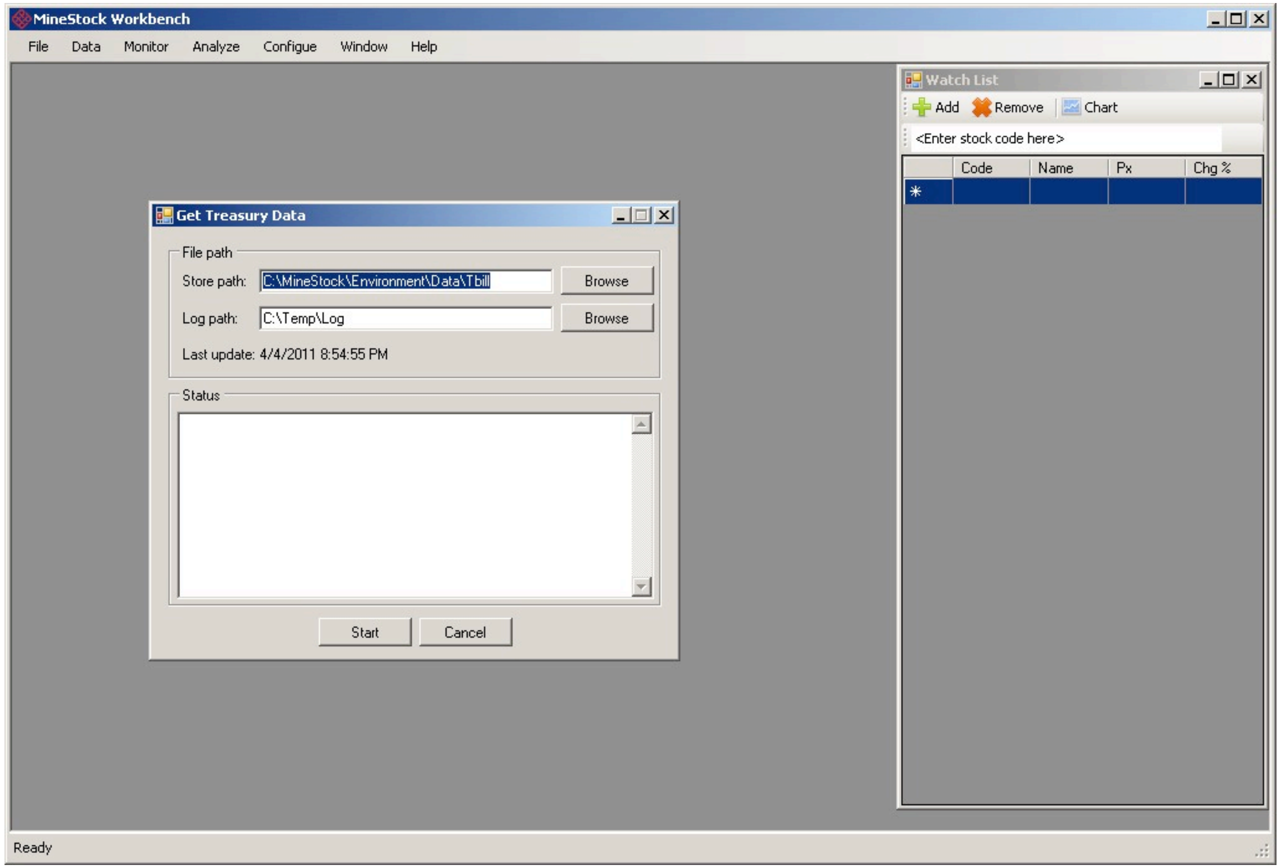
File path

Output path: C:\MineStock\Environment\Data\Index

Log path: C:\Temp\Log

Status

Figure 19: Downloading stocks and index price in MineStock Workbench

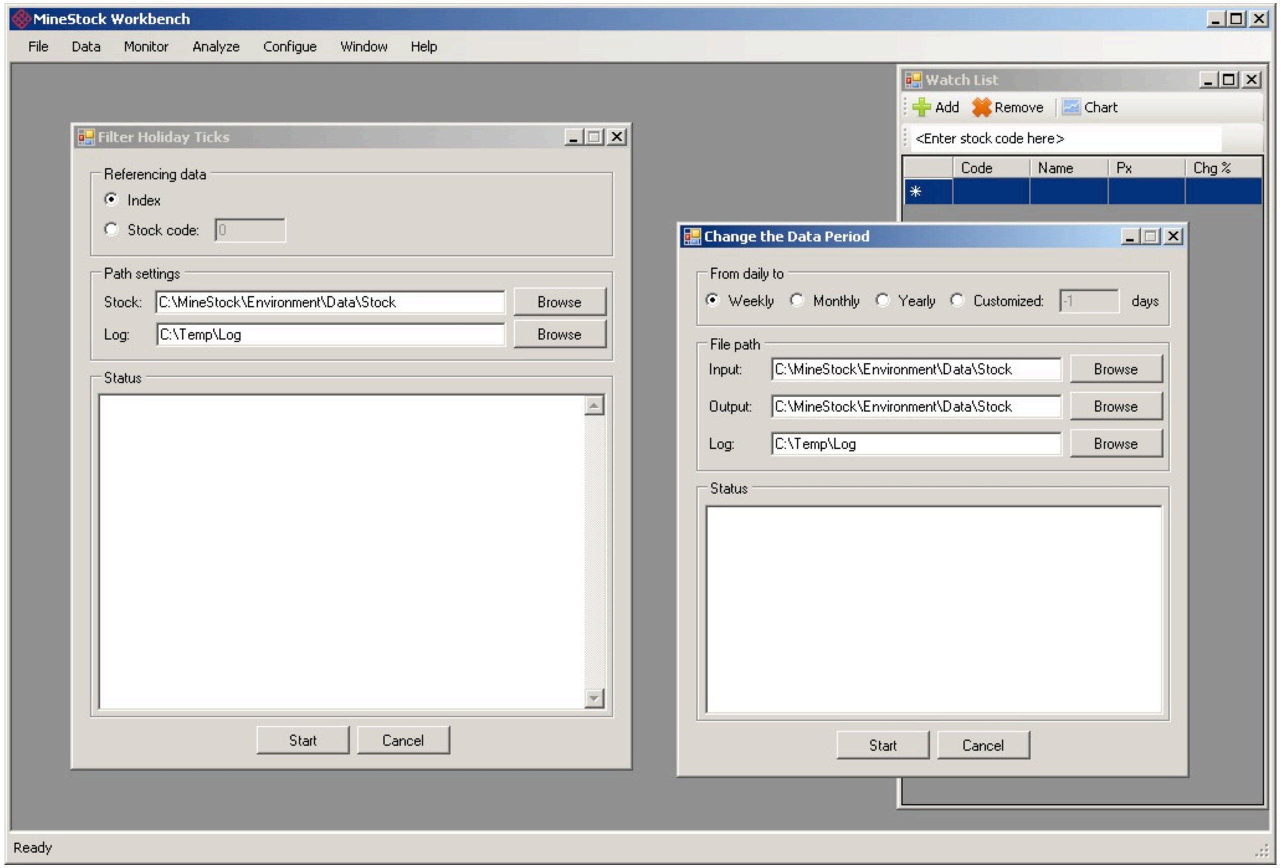


Store path: C:\MineStock\Environment\Data\Tbill

Log path: C:\Temp\Log

Last update: 4/4/2011 8:54:55 PM

Figure 20: Downloading treasury data in MineStock Workbench



Filter Holiday Ticks

Referencing data

● Index

○ Stock code: 0

Path settings

Stock: C:\MineStock\Environment\Data\Stock

Log: C:\Temp\Log

Status

Watch List

<Enter stock code here>

Change the Data Period

From daily to

● Weekly ○ Monthly ○ Yearly ○ Customized: -1 days

File path

Input: C:\MineStock\Environment\Data\Stock

Output: C:\MineStock\Environment\Data\Stock

Log: C:\Temp\Log

Status

Figure 21: Filtering holiday ticks and changing data’s time interval in MineStock Workbench

7.2.4 Compile Statistics and Performance Indicators

A set of statistics and performance indicators of stock listed in section 5.3 are computed by this function in the system. To access the function, users can select 'Data', 'Preprocess Data', then 'Compile Statistics...' to open the panel. A panel same as the one in figure 22 will be shown. In the panel, inputs including the system paths for storage of stocks, index, treasury bill, log files, also the output path for calculation results are needed. After clicking 'Start', the system will start to compute the indicators.

Note that the stocks and index files given to this function have to be in the same period and interval. Otherwise, the results calculated would be inaccurate.

7.2.5 Discretize Data by Price Change

As discussed in section 5.2.5, discretization of price data of stocks is needed for 2 of the 3 clustering algorithms currently offered by the system. To discretize the stock price data according to their magnitude of changes in each day, users can select 'Data', 'Preprocess Data', then 'Discretize Data by Price Change...'. A panel same as the one in figure 23 will be shown. In the panel, inputs including the storage path of stock data, output path of discretized result of stock prices, namely the 'Genus'[4], and also the logging path are needed. Most importantly, the number of 'Genus' to be separated for the set the stock prices, is also need to be defined. After clicking the 'Start' button, the system will start to perform discretization on the stocks provided and store the results to the 'Genus' path.

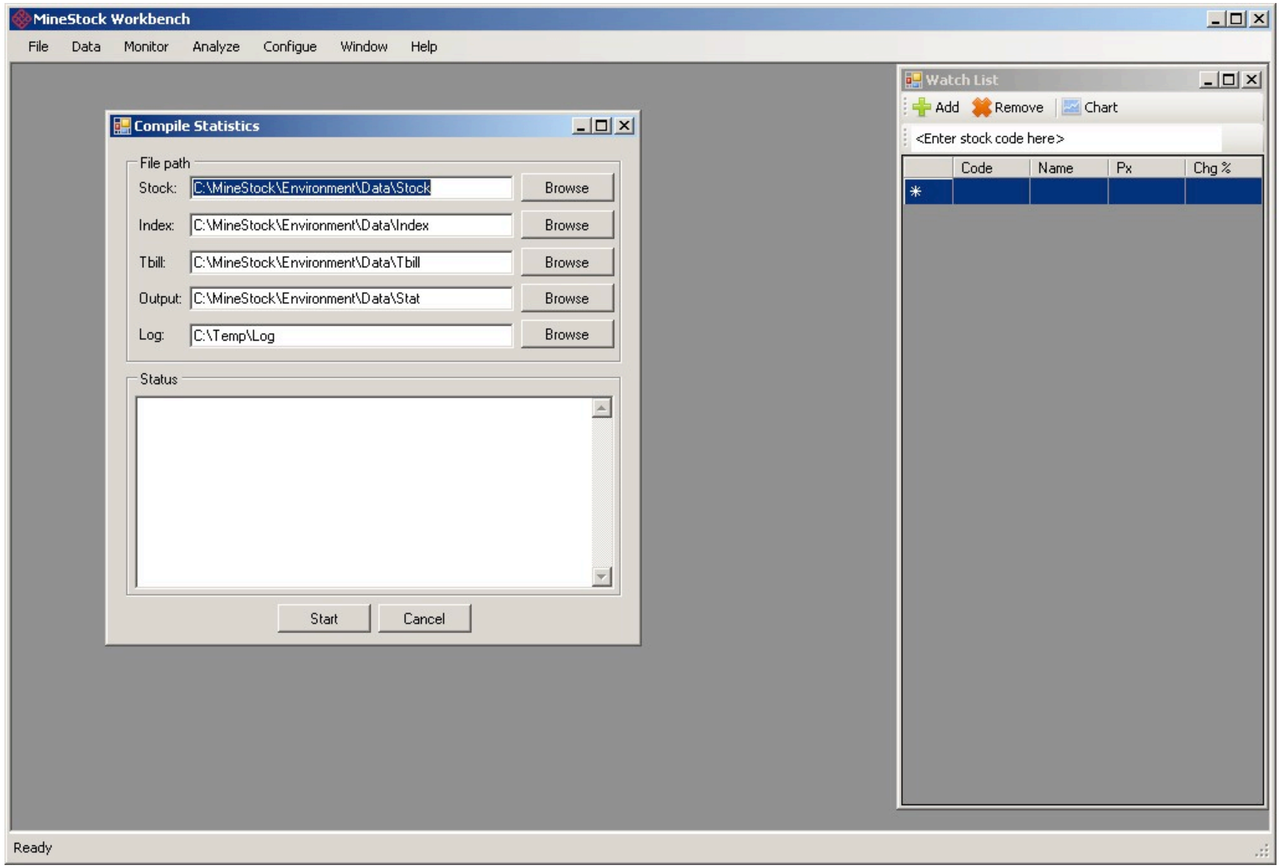
7.2.6 Discretize Data by Price Level

To discretize the stock price data according to their magnitude of price deviation from the mean of the price's referencing period in each day, users can select 'Data', 'Preprocess Data', then 'Discretize Data by Price Level...'. A panel same as the one in figure 24 will be shown. In the panel, inputs including the storage path of stock data, output path of discretized result of stock prices, namely the 'Genus', and also the logging path are needed. Most importantly, the number of 'Genus' to be separated for the set the stock prices and the number of calendar days in a referencing period are also needed to be defined. After clicking 'Start', the system will start to perform discretization on the stocks provided and store the results to the 'Genus' path. More details regarding the referencing period can refer to section 5.2.5.

7.3 Monitoring Functions

The system provides several monitoring functions on stocks including the customizable watch list, portfolio, and also the ability of viewing stock charts.

[4] The term 'Genus' is used across the whole system to represent the discretized result of stock prices, a separation of stock prices according to their happened nature.



Stock: C:\MineStock\Environment\Data\Stock

Index: C:\MineStock\Environment\Data\Index

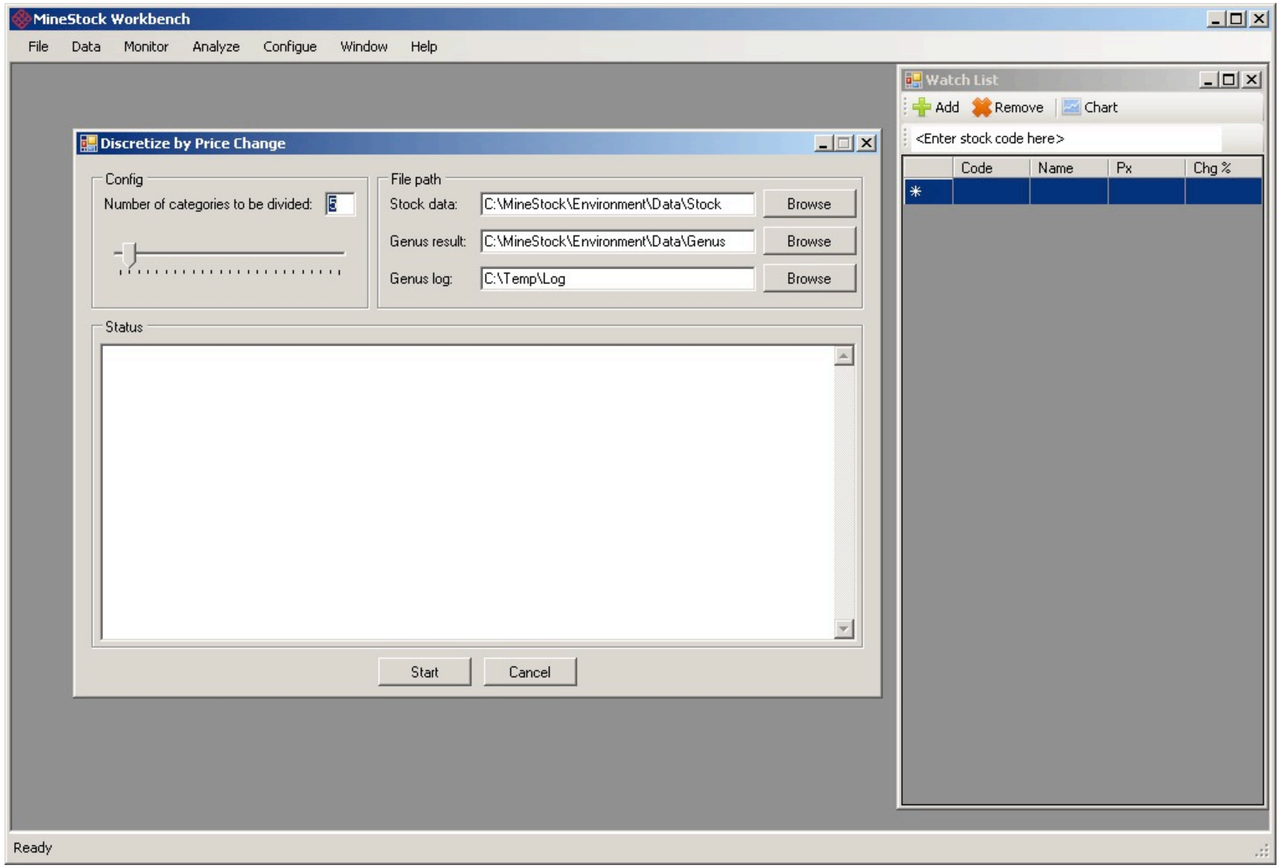
Tbill: C:\MineStock\Environment\Data\Tbill

Output: C:\MineStock\Environment\Data\Stat

Log: C:\Temp\Log

<Enter stock code here>

Figure 22: Compiling statistics in MineStock Workbench



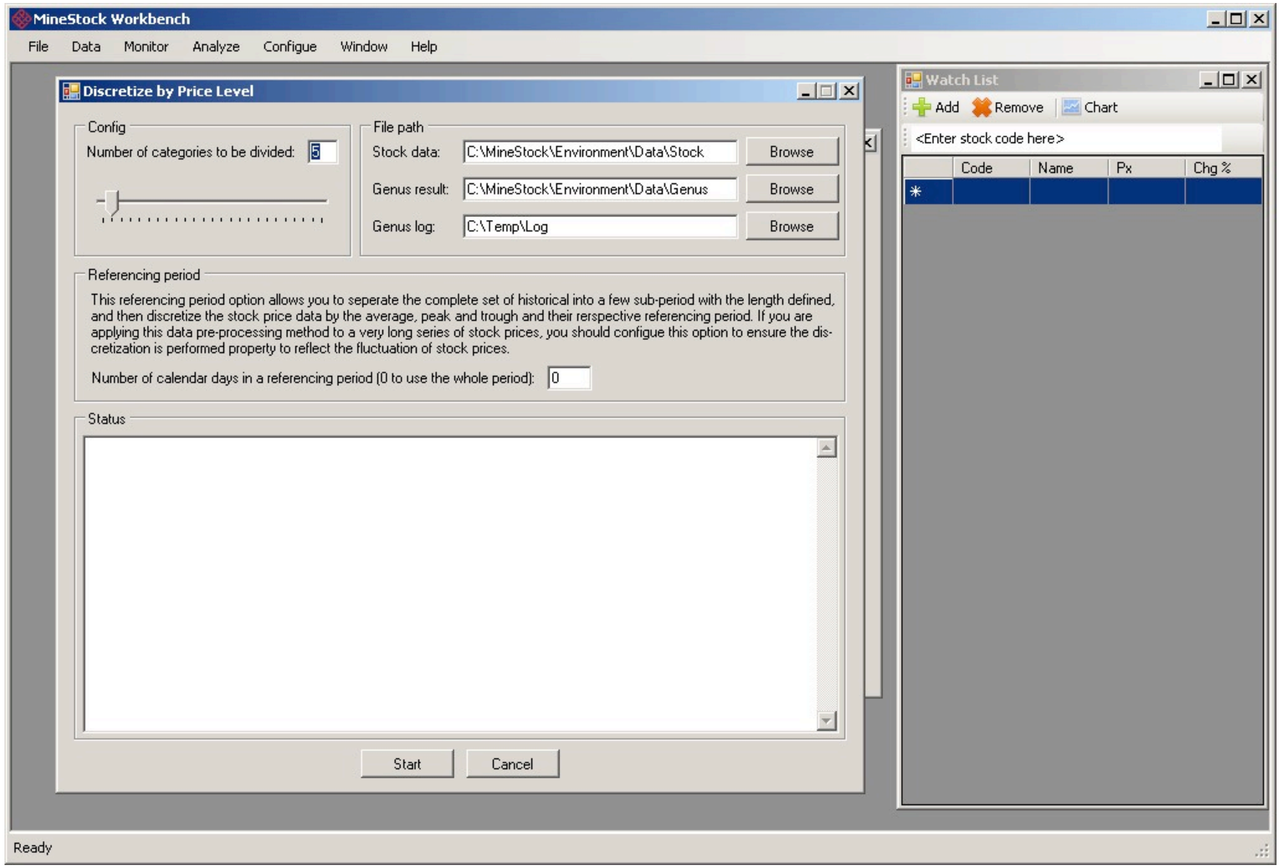
Number of categories to be divided: 5

Stock data: C:\MineStock\Environment\Data\Stock

Genus result: C:\MineStock\Environment\Data\Genus

Genus log: C:\Temp\Log

Figure 23: Discretizing data by price change in MineStock Workbench



Config

Number of categories to be divided: 5

File path

Stock data: C:\MineStock\Environment\Data\Stock

Genus result: C:\MineStock\Environment\Data\Genus

Genus log: C:\Temp\Log

Referencing period

This referencing period option allows you to seperate the complete set of historical into a few sub-period with the length defined, and then discretize the stock price data by the average, peak and trough and their rerspective referencing period. If you are applying this data pre-processing method to a very long series of stock prices, you should configue this option to ensure the discretization is performed property to reflect the fluctuation of stock prices.

Number of calendar days in a referencing period (0 to use the whole period): 0

Status

Figure 24: Discretizing data by price level in MineStock Workbench

7.3.1 Watch List and Portfolio

The customizable watch list will be automatically opened each time when the system has been started up. Alternatively, users can access the watch list by selecting 'Monitor', and then 'Watch List'. A sample of watch list can refer to the right hand side of figure 25. After adding the stocks by typing their code in the upper text box of the panel, or directly in the 'Code' column of the grid view, the columns of stock name, latest price, and the latest percentage change will be refreshed automatically whenever the latest stock price has been updated by the price downloading function described in section 7.2.1. Removing stocks in the watch list is also supported by clicking the 'Remove' button on the top of the panel.

Users of the system can also submit their current portfolio in hand to the system, therefore, monitor the performance of the individual stocks inside it using the portfolio function. To access it, users can select 'Monitor', and then 'Portfolio'. A panel same as the one in the left side of figure 25 will then be shown. Similar to the watch list, users can add stocks to the portfolio by typing their code in the upper text box of the panel, or directly in the 'Code' column of the grid view, and also remove stocks by selecting a row in the gird view and clicking the 'Remove' button.

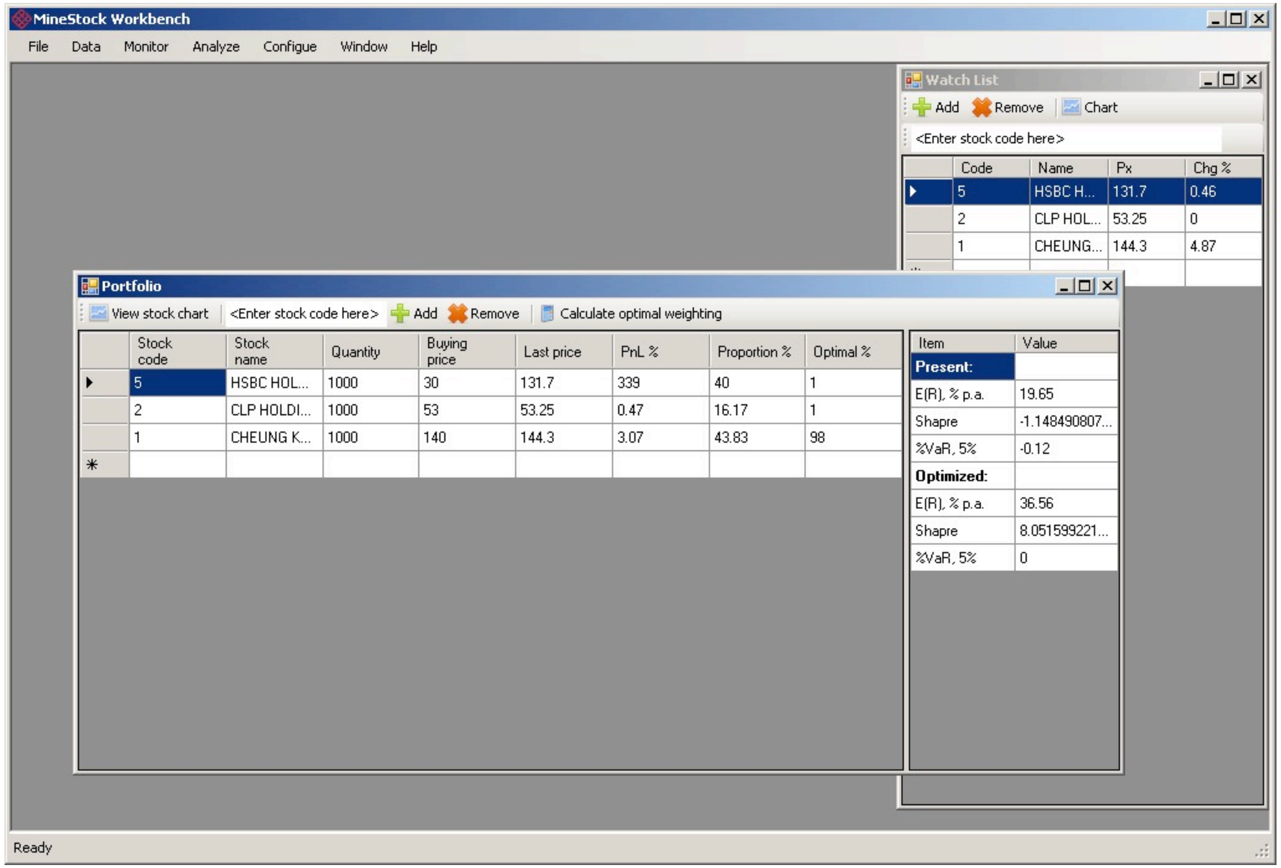
The unrealized profit and loss (PnL) percentage will be calculated automatically if the users have inputted the quantity purchased and price of purchase of their stocks. It will also be automatically refreshed whenever the latest stock price has been updated by the price downloading function. In the right section inside the portfolio panel, under 'Present' heading, the average annual return and Sharpe ratio of the portfolio are shown.

Every change in portfolio will be immediately saved into the environment path of system.

7.3.2 Optimize portfolio Weightings

The column 'Optimal %' in the gird view of figure 25 shows the optimal weighting for the stocks in the portfolio. Also, in the right section inside the portfolio panel, under 'Optimized' heading, the average return per annum and Sharpe ratio of the optimized portfolio are shown. To calculate these optimized figures, users can click 'Calculate optimal weighting' button on the top of the portfolio panel. A dialog box same as the one in figure 26 will be shown. In the dialog box, several inputs are needed including the environment path which stores the portfolio information, paths of log files, stocks, treasury bills, and also the path of statistic results of stocks calculated with the compile statistics function discussed in section 7.2.4.

After clicking start, the system will initiate its process to find out the optimal weighting of each stock in the portfolio. When the process is finished, the above mentioned metrics of optimal portfolio will be refreshed instantly.

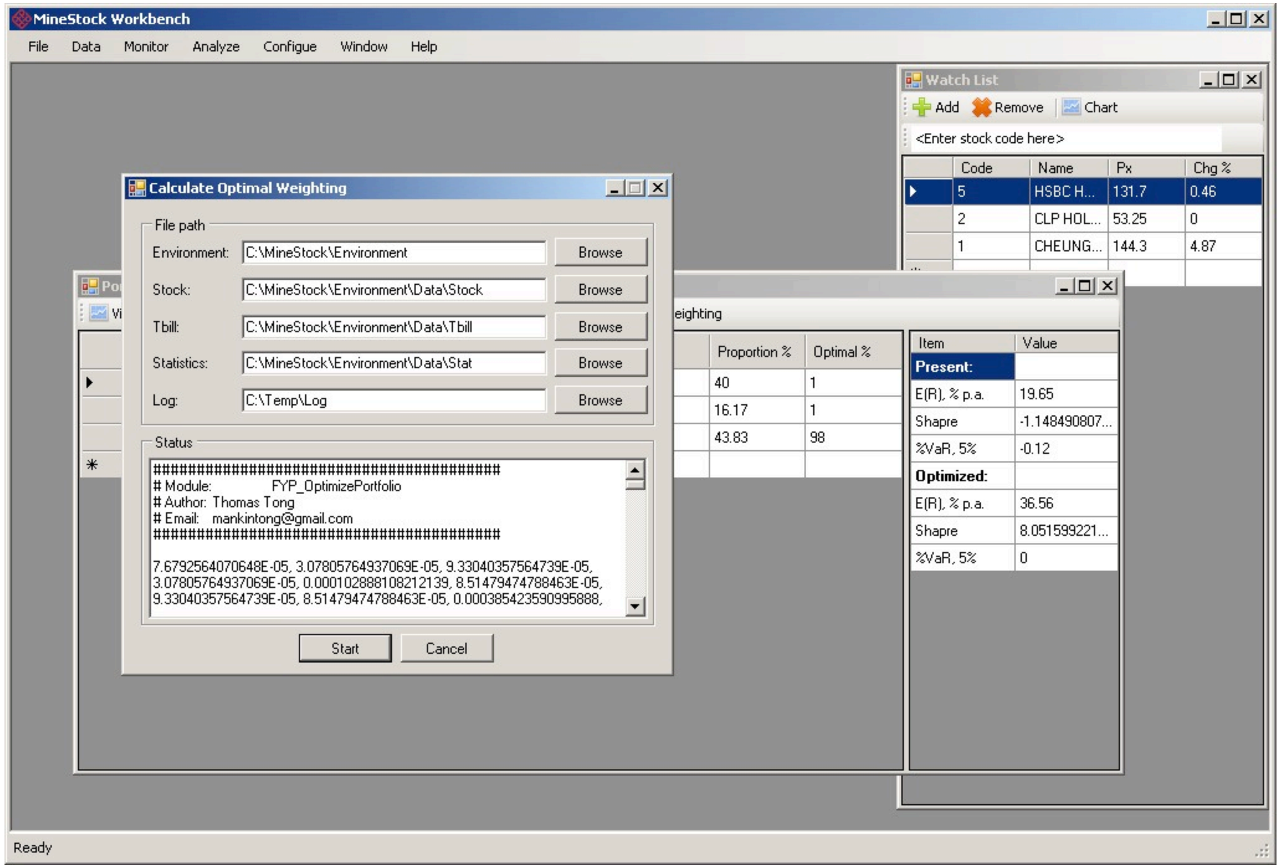


Code	Name	Px	Chg %
5	HSBC H...	131.7	0.46
2	CLP HOL...	53.25	0
1	CHEUNG...	144.3	4.87

Stock code	Stock name	Quantity	Buying price	Last price	PnL %	Proportion %	Optimal %
5	HSBC HOL...	1000	30	131.7	339	40	1
2	CLP HOLDI...	1000	53	53.25	0.47	16.17	1
1	CHEUNG K...	1000	140	144.3	3.07	43.83	98

Item	Value
Present:	
E(R), % p.a.	19.65
Shapre	-1.148490807...
%VaR, 5%	-0.12
Optimized:	
E(R), % p.a.	36.56
Shapre	8.051599221...
%VaR, 5%	0

Figure 25: Managing portfolio and watch list in MineStock Workbench



<Enter stock code here>

Code	Name	Px	Chg %
5	HSBC H...	131.7	0.46
2	CLP HOL...	53.25	0
1	CHEUNG...	144.3	4.87

File path

Environment: C:\MineStock\Environment

Stock: C:\MineStock\Environment\Data\Stock

Tbill: C:\MineStock\Environment\Data\Tbill

Statistics: C:\MineStock\Environment\Data\Stat

Log: C:\Temp\Log

Status

```
#########################################
# Module:              FYP_OptimizePortfolio
# Author:  Thomas Tong
# Email:   mankintong@gmail.com
#########################################

7.6792564070648E-05, 3.07805764937069E-05, 9.33040357564739E-05,
3.07805764937069E-05, 0.000102888108212139, 8.51479474788463E-05,
9.33040357564739E-05, 8.51479474788463E-05, 0.000385423590995888,
```

eighting

Proportion %	Optimal %
40	1
16.17	1
43.83	98

Item	Value
Present:	
E(R), % p.a.	19.65
Shapre	-1.148490807...
%VaR, 5%	-0.12
Optimized:	
E(R), % p.a.	36.56
Shapre	8.051599221...
%VaR, 5%	0

Figure 26: Calculating optimal portfolio weightings in MineStock Workbench

7.3.3 Stock Charts

There are many ways for users of the system to view the chart of stocks. They can select the stock in the watch list and then click the ‘Chart’ button, or select the stock in the portfolio panel and then click the ‘View stock chart’ button. Alternatively, they can also select ‘Monitor’ in the menu, and then ‘View Chart...’. A dialog box same as the one in left side of figure 27 will be shown.

With this dialog box, users can customize lines of stocks and compare stocks by defining the storage path of stock data files, and then selecting their desired stock code and color, then clicking the ‘Add’ button. When they have selected the stocks they want to view in chart, they can click the ‘Chart’ button to view it, as shown in the right hand side of figure 27.

7.4 Analyzing Functions

7.4.1 Classical K-means Technique

To perform k-means clustering on stocks, users can select ‘Analyze’, then ‘K-means Clustering Method...’ to open the panel. A panel same as the one in figure 28 will be shown. In the panel, several inputs are needed, including the storage path of stock data, cluster data, and log files, and also the number of clusters to be divided. After clicking the ‘Start’ button, the system will initiate the process of k-means clustering algorithm to handle the given stocks.

Section 5.2.1 can be referred for more details of this algorithm.

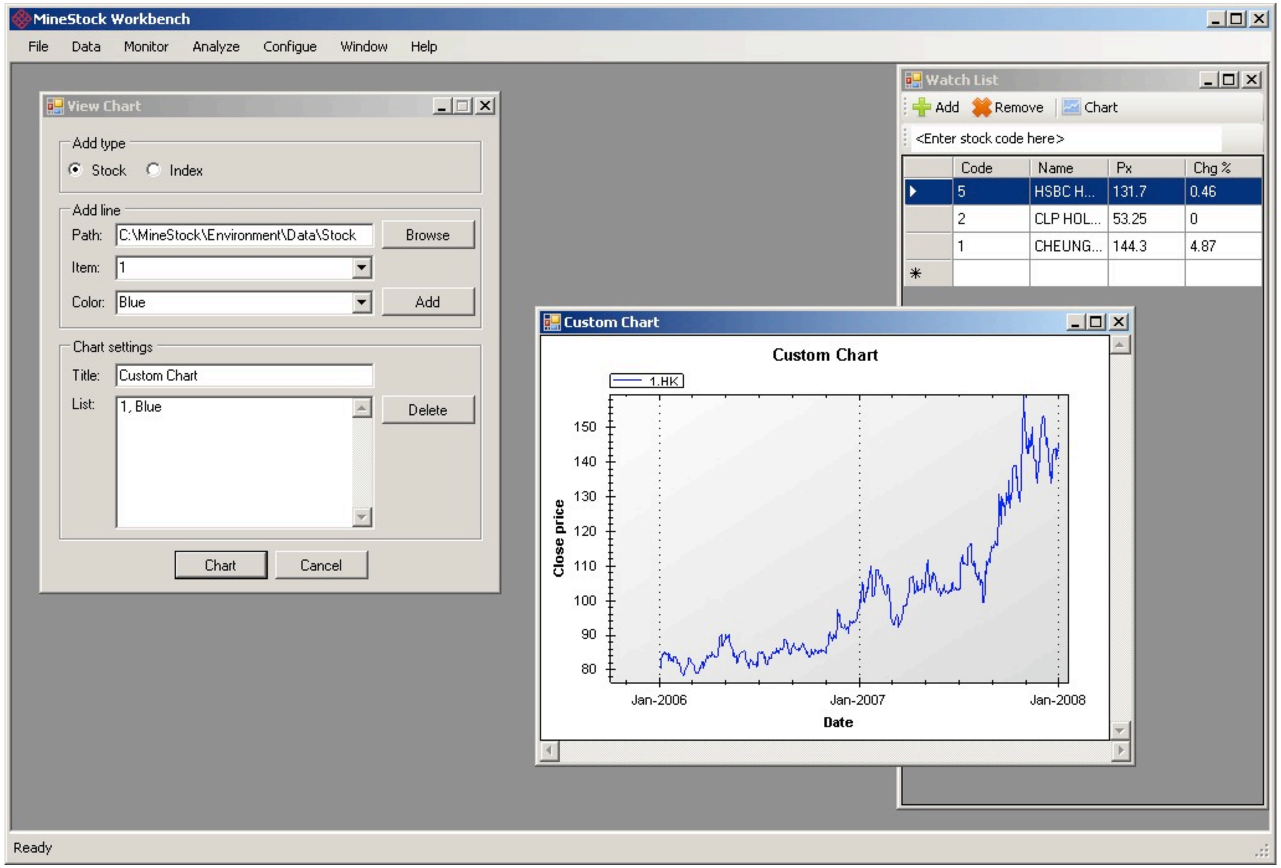
7.4.2 Identical Sequence Extraction

To perform sequence extraction clustering on stocks, users can select ‘Analyze’, then ‘Identical Fixed Length Sequence Discovery...’ to open the panel. A panel same as the one in figure 29 will be shown. In the panel, several inputs are needed. They include the storage path of stock data, cluster data, and log files, and also the number of clusters to be divided and the length of the windows. After clicking the ‘Start’ button, the system will initiate the process of identical sequence extraction clustering algorithm to handle the given stocks.

To know more about this algorithm and the use of these parameters, please refer to section 5.2.2.

7.4.3 Similar Motif Discovery

To perform motif discovery clustering on stocks, users can select ‘Analyze’, then ‘Similar Motif Discovery...’ to open the panel. A panel same as the one in figure 30 will be shown. In the panel, several inputs are needed. They include the storage path of stock data, cluster data, and log files, and also the number of clusters to be divided and the value of similarity multiplier. After clicking the



Add type

● Stock ○ Index

Add line

Path: C:\MineStock\Environment\Data\Stock

Item: 1

Color: Blue

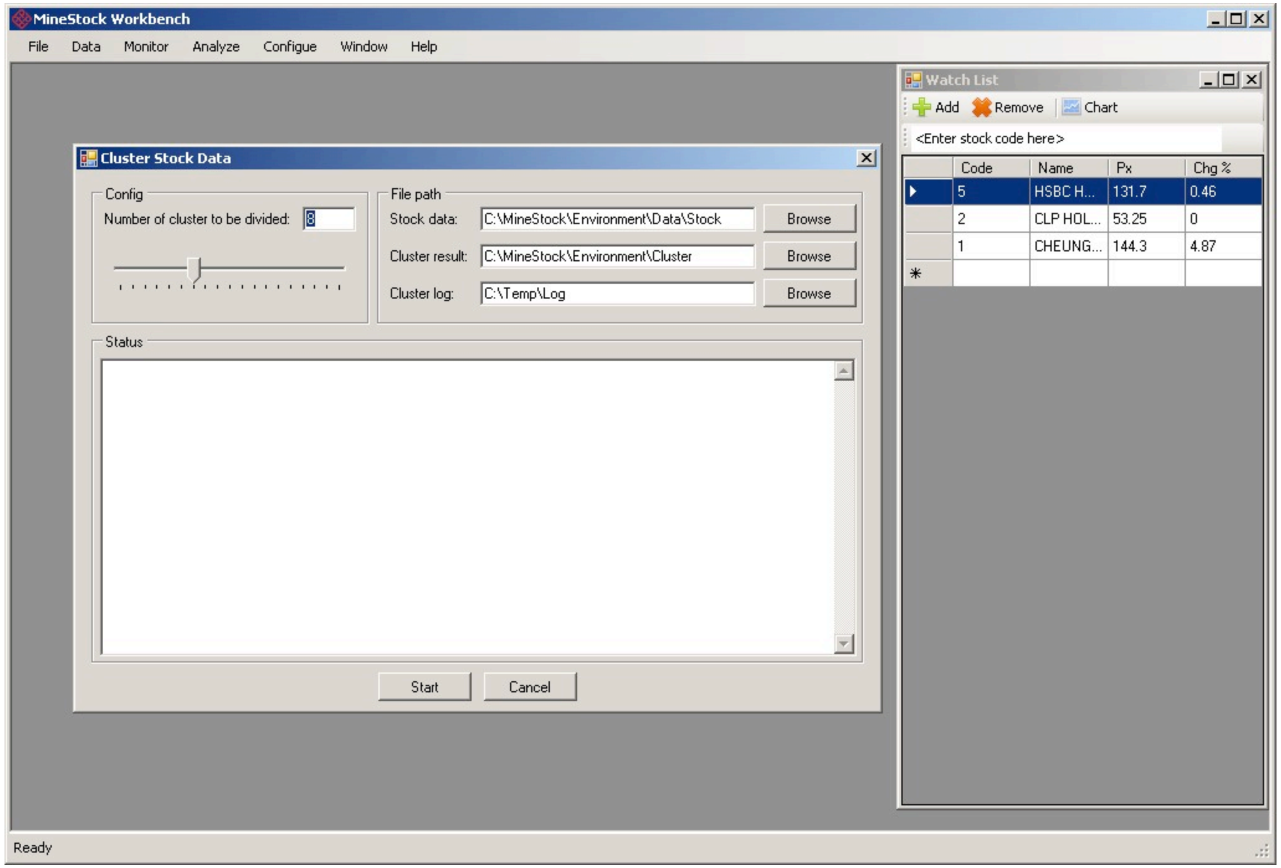
Chart settings

Title: Custom Chart

List: 1, Blue

	Code	Name	Px	Chg %
▶	5	HSBC H...	131.7	0.46
	2	CLP HOL...	53.25	0
	1	CHEUNG...	144.3	4.87
*				

Figure 27: Viewing stock charts in MineStock Workbench



Number of cluster to be divided: 8

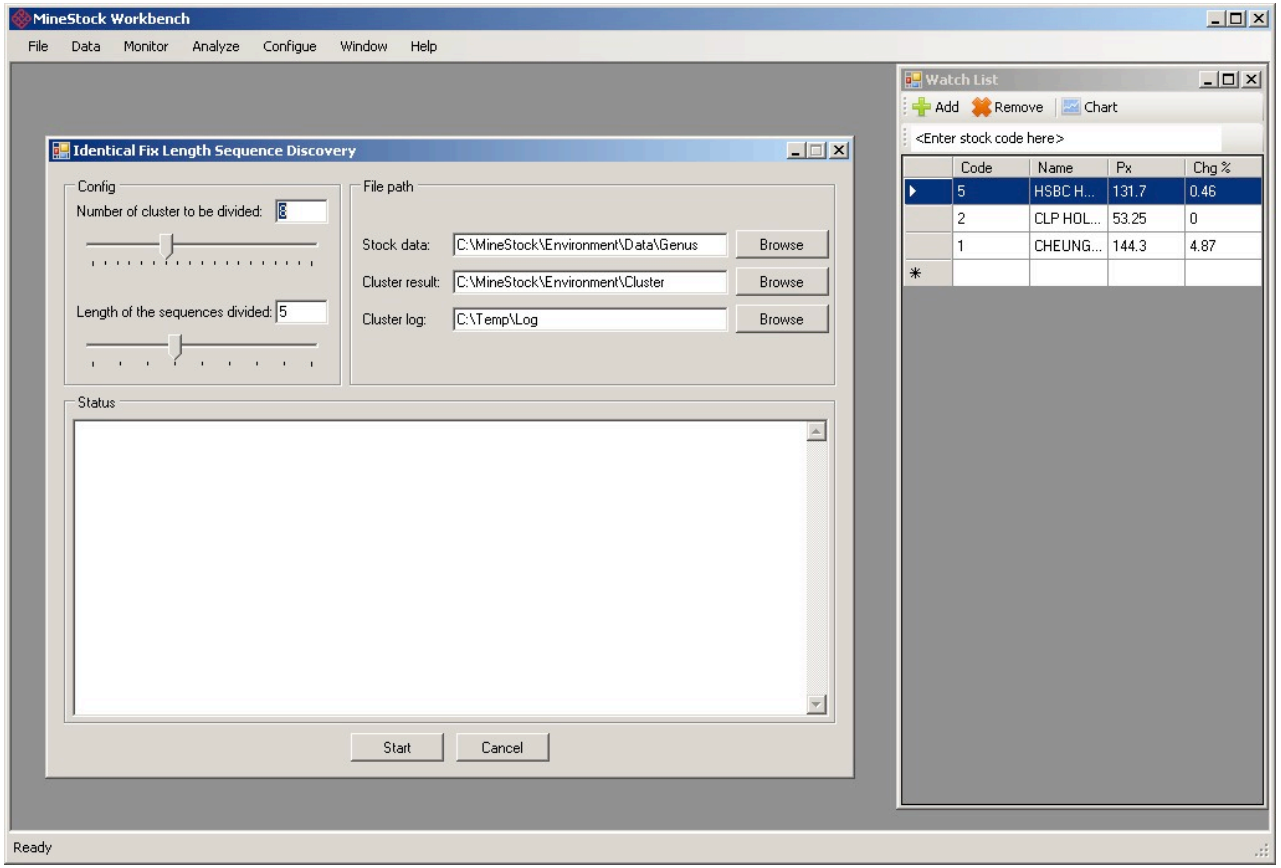
Stock data: C:\MineStock\Environment\Data\Stock

Cluster result: C:\MineStock\Environment\Cluster

Cluster log: C:\Temp\Log

	Code	Name	Px	Chg %
▶	5	HSBC H...	131.7	0.46
	2	CLP HOL...	53.25	0
	1	CHEUNG...	144.3	4.87
✳				

Figure 28: Clustering by k-means in MineStock Workbench



Number of cluster to be divided: 8

Length of the sequences divided: 5

Stock data: C:\MineStock\Environment\Data\Genus

Cluster result: C:\MineStock\Environment\Cluster

Cluster log: C:\Temp\Log

<Enter stock code here>

	Code	Name	Px	Chg %
▶	5	HSBC H...	131.7	0.46
	2	CLP HOL...	53.25	0
	1	CHEUNG...	144.3	4.87
*				

Figure 29: Clustering by identical sequence extraction in MineStock Workbench

‘Start’ button, the system will initiate the process of similar motif discovery clustering algorithm to handle the given stocks.

Readers can refer to section 5.2.3 for more about this algorithm and the use of the above mentioned parameters.

7.4.4 View Results

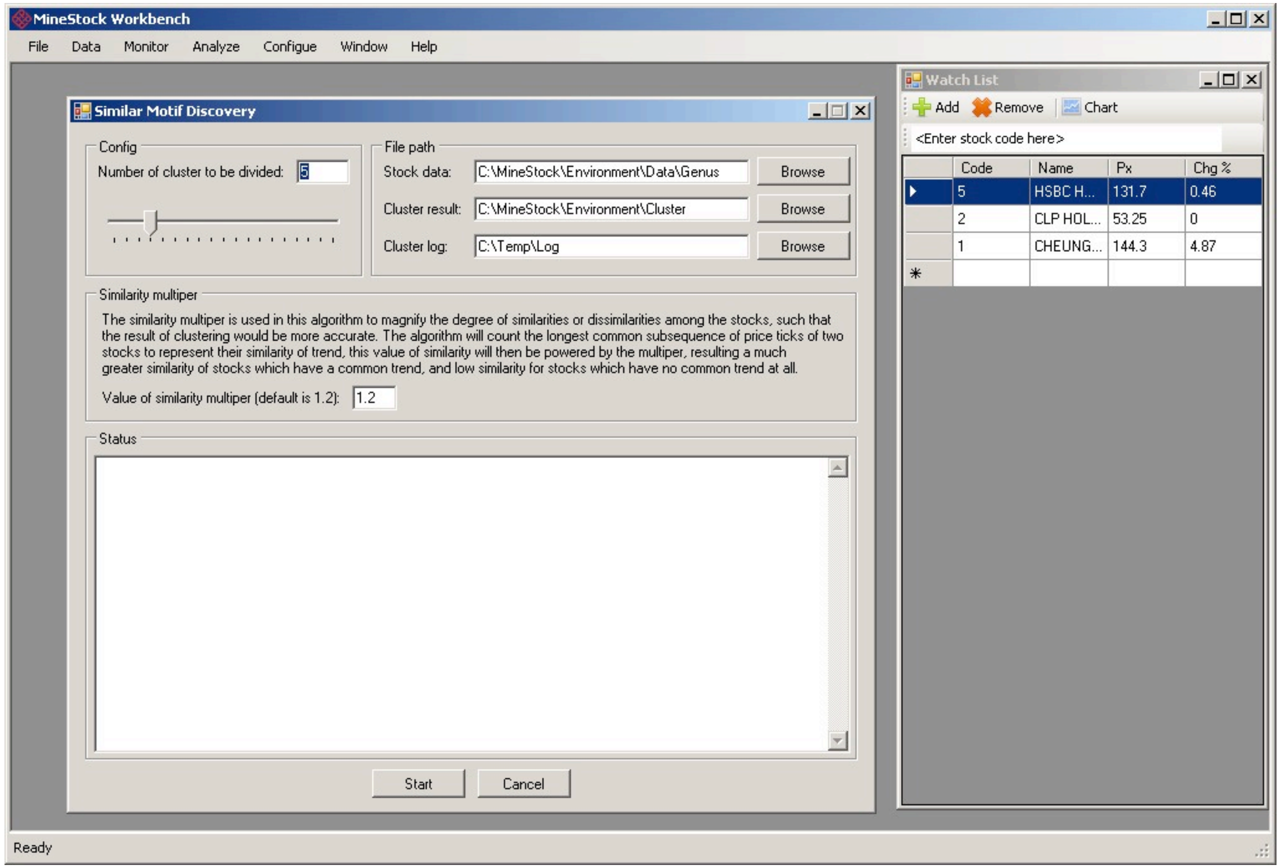
After the clustering processes or the calculation of performance indicators have been done. Users can access the panel shown in figure 31 to view the results by clicking ‘Analyze’ in the menu, and then ‘View Results’. In the left hand side of the ‘View Results’ panel, it is the tree view of clusters found under the cluster path of the system environment directory. In the figure we can see that another set of sub-clusters are located under ‘Cluster 1’ located in ’MOTIF’ folder. This implies that an extra iteration of clustering has been done on that ‘Cluster 1’.

Clicking any clusters in the left hand side of the ‘View Results’ panel, the list of stocks assigned for this particular cluster, together with their results on different performance indicators are shown in the right hand side of the screen. For more information regarding these indicators, please refer to section 5.3. The data grid can be sorted in both ascending and descending orders by clicking on the header of column which investors wish to sort with.

As shown in figure 31, 2 filter can be applied to the data grid of stocks. The first one is aimed to filter the stocks which are having insufficient price data and is particularly useful when the data grid is set to display all the available stocks. It is because some stocks have been traded for just a short time, therefore, in the sampling period defined, the historical price ticks acquired for those stocks may be significantly less than other stocks. Performance indicators based on a small number of price ticks may not be accurate at all and a filter can help us to eliminate this problem. Under this function, if the number of price ticks of a stock is less than 80 percent of other stocks, the stock will be filtered from the grid view.

The second filter is aimed to filter the stocks which are significantly underperformed. These stocks are, in theory, not recommended to invest no matter which clusters they are in. The writer defines the stocks which are having a negative Sharpe ratio, that is, having a even lower return compared with risk-free instrument, as the stocks to be filtered by this function. This function enables users to shortlist the stocks which are having better performance in each cluster, they are, therefore, easier to select the stocks for their portfolio.

The result panel can also be linked with the portfolio panel to provide a more integrated environment for investors to select the appropriate stocks. To do this, users can open the portfolio screen and then click the ‘Link portfolio’ button on the top of the result panel. After they are linked, the clusters which are having stocks in the portfolio will be highlighted, so investors can focus on selecting stocks from other clusters in order to achieve a greater diversification effect.



Number of cluster to be divided: 5

Stock data: C:\MineStock\Environment\Data\Genus

Cluster result: C:\MineStock\Environment\Cluster

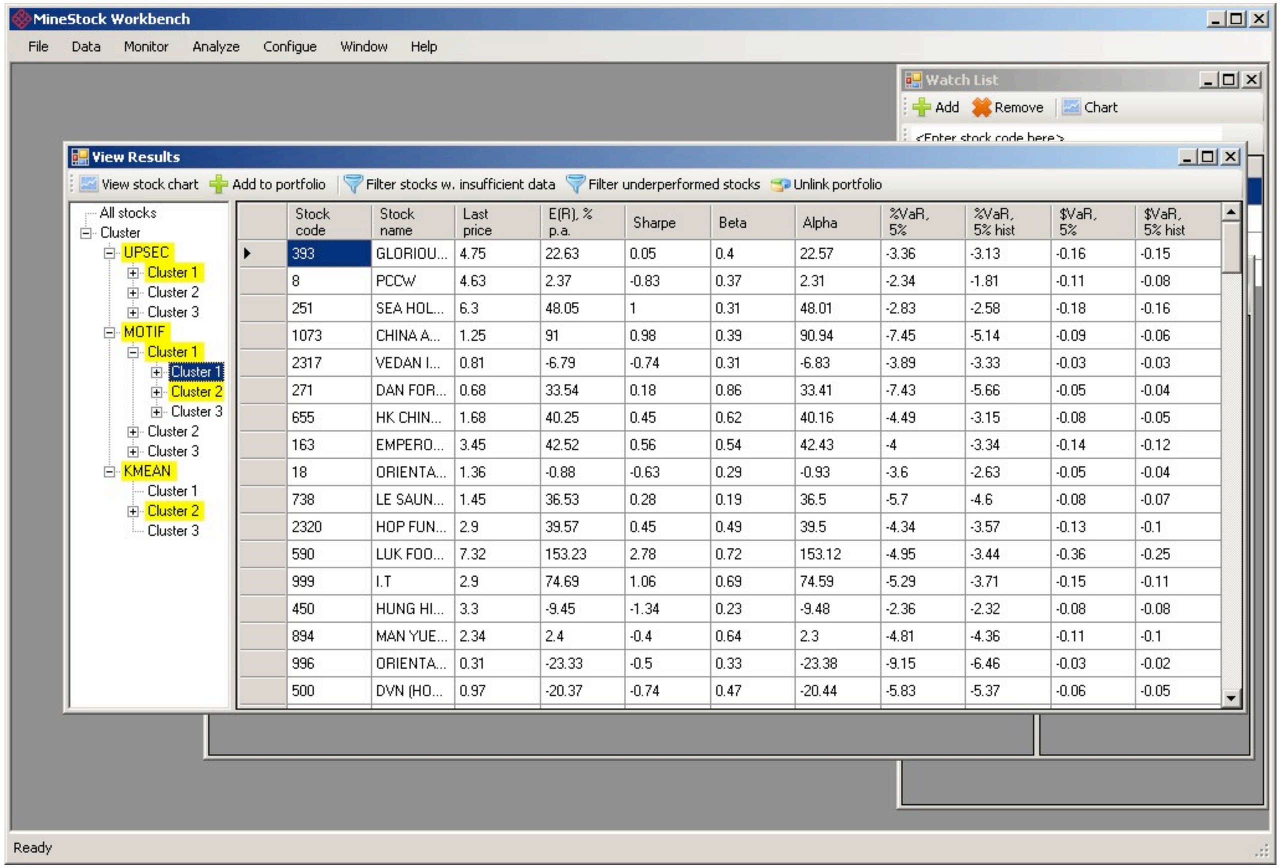
Cluster log: C:\Temp\Log

The similarity multiper is used in this algorithm to magnify the degree of similarities or dissimilarities among the stocks, such that the result of clustering would be more accurate. The algorithm will count the longest common subsequence of price ticks of two stocks to represent their similarity of trend, this value of similarity will then be powered by the multiper, resulting a much greater similarity of stocks which have a common trend, and low similarity for stocks which have no common trend at all.

Value of similarity multiper (default is 1.2): 1.2

Code	Name	Px	Chg %
5	HSBC H...	131.7	0.46
2	CLP HOL...	53.25	0
1	CHEUNG...	144.3	4.87

Figure 30: Clustering by similar motif discovery in MineStock Workbench



Stock code	Stock name	Last price	E(R), % p.a.	Sharpe	Beta	Alpha	%VaR, 5%	%VaR, 5% hist	$VaR, 5%	$VaR, 5% hist
393	GLORIOU...	4.75	22.63	0.05	0.4	22.57	-3.36	-3.13	-0.16	-0.15
8	PCCW	4.63	2.37	-0.83	0.37	2.31	-2.34	-1.81	-0.11	-0.08
251	SEA HOL...	6.3	48.05	1	0.31	48.01	-2.83	-2.58	-0.18	-0.16
1073	CHINA A...	1.25	91	0.98	0.39	90.94	-7.45	-5.14	-0.09	-0.06
2317	VEDAN I...	0.81	-6.79	-0.74	0.31	-6.83	-3.89	-3.33	-0.03	-0.03
271	DAN FOR...	0.68	33.54	0.18	0.86	33.41	-7.43	-5.66	-0.05	-0.04
655	HK CHIN...	1.68	40.25	0.45	0.62	40.16	-4.49	-3.15	-0.08	-0.05
163	EMPERO...	3.45	42.52	0.56	0.54	42.43	-4	-3.34	-0.14	-0.12
18	ORIENTA...	1.36	-0.88	-0.63	0.29	-0.93	-3.6	-2.63	-0.05	-0.04
738	LE SAUN...	1.45	36.53	0.28	0.19	36.5	-5.7	-4.6	-0.08	-0.07
2320	HOP FUN...	2.9	39.57	0.45	0.49	39.5	-4.34	-3.57	-0.13	-0.1
590	LUK FOO...	7.32	153.23	2.78	0.72	153.12	-4.95	-3.44	-0.36	-0.25
999	I.T	2.9	74.69	1.06	0.69	74.59	-5.29	-3.71	-0.15	-0.11
450	HUNG HI...	3.3	-9.45	-1.34	0.23	-9.48	-2.36	-2.32	-0.08	-0.08
894	MAN YUE...	2.34	2.4	-0.4	0.64	2.3	-4.81	-4.36	-0.11	-0.1
996	ORIENTA...	0.31	-23.33	-0.5	0.33	-23.38	-9.15	-6.46	-0.03	-0.02
500	DVN (HO...	0.97	-20.37	-0.74	0.47	-20.44	-5.83	-5.37	-0.06	-0.05

Figure 31: Viewing statistical and clustering results of stocks in MineStock Workbench

7.4.5 Export Results

While all the system files are stored in XML format as discussed in section 6.3, there is an export function which allows users to export the system's data in CSV format. To access this function, users can click 'File' in the top menu, then 'Export...'. A dialog box will then pop up, as shown in figure 32, for users to select the type and the data file that they would like to export.

7.4.6 Define Batch Actions

In the previous sections, we have discussed many functions provided by the system and the way to access them individually. However, accessing the functions one by one may not be the best way if the users wish to make use of multiple functions together for a more complex analysis. The 'Automator' function provided by the system is to make complex studies easier by letting the advance users to develop an automation script. The scripts are also supported to be saved as files, so that users of this system may exchange their scripts, and thus, also exchange their ideas and parameters used for their studies.

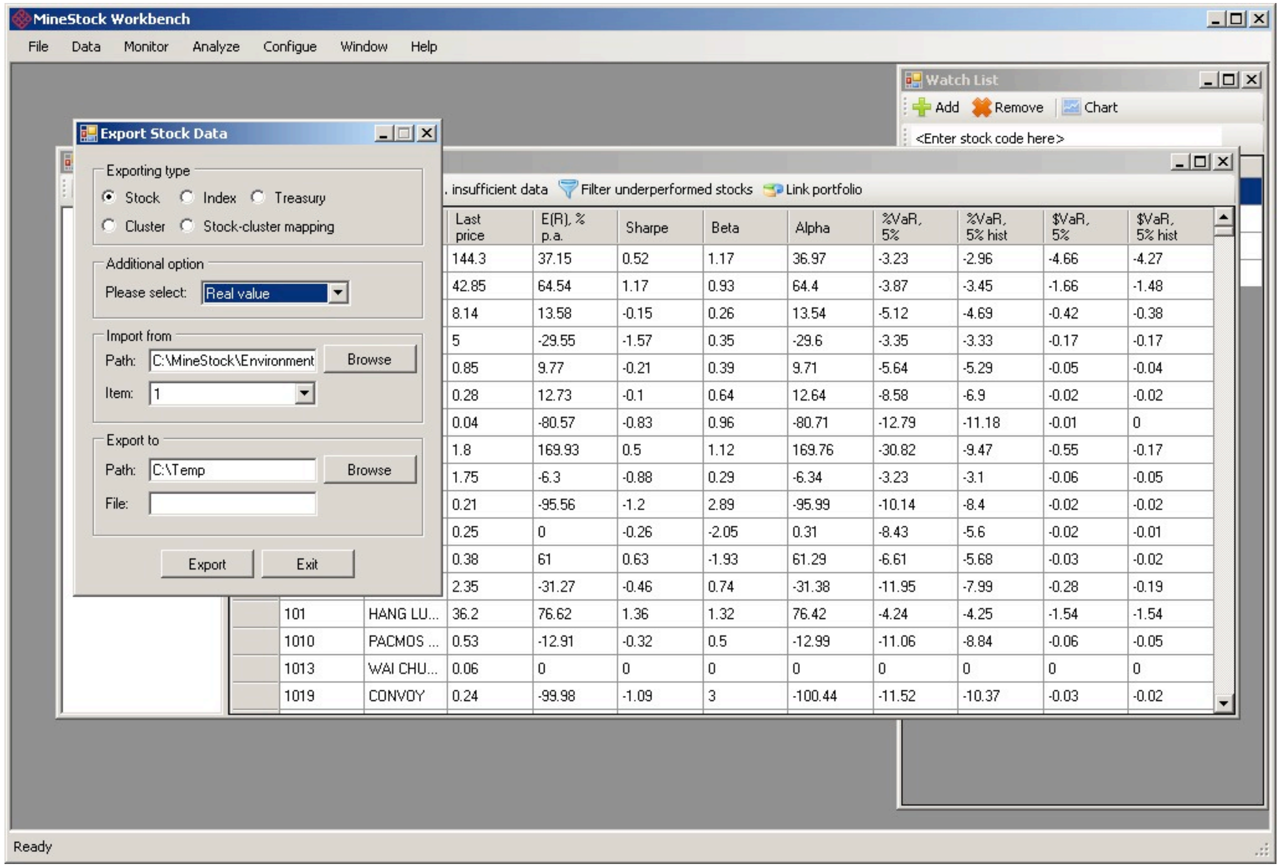
To access the automator function, users can select 'Analyze' in the menu, and then 'Automator'. A panel same as the one in figure 33 will be shown. In the panel, users can select the functions they would like to use from the drop down list on the top, and then clicking the 'Add' button. A new panel will then pop up and ask for the parameters for the function as they are accessed in the normal ways described in above sections. After the parameters needed by the function have been inputted, a new row of action will be added in the data grid of the automator panel. Users can also remove and modify the settings of each action added into the data grid by clicking 'Remove' and 'Change selected settings' button. The order of each new action added is defined the be the last by default, but these order in sequence of each action can be reassigned by selecting the action and then clicking the 'Move up' and 'Move down' button.

When investors defined the list automation actions using the panel, they can save the list as external file by clicking 'Save' button and load them at a later time using the 'Load' button. For both buttons, after they are clicked, a file chooser will be shown for letting the users to select the folder and file.

7.4.7 Define Batch Actions on Subset

In methodologies section 5.2.4, we have discussed the possibilities of performing further clustering on the clusters we have. While the investors can only execute the algorithms one by one with changing the input and output paths manually, the automator also simplify this process by introducing the capabilities to perform certain actions on only subset of stocks. The subset will be populated according to the user defined criteria, for instance, only the stocks inside a particular cluster with more than 5 stocks.

To define batch actions on subset of stocks, investors can make use of the buttons 'Use subset for following action' and 'Use universal set' located on the top of the automator screen as shown in figure 33. Clicking 'Use subset for following



Export Stock Data

Exporting type

● Stock ○ Index ○ Treasury

○ Cluster ○ Stock-cluster mapping

Additional option

Please select: Real value

Import from

Path: C:\MineStock\Environment

Item: 1

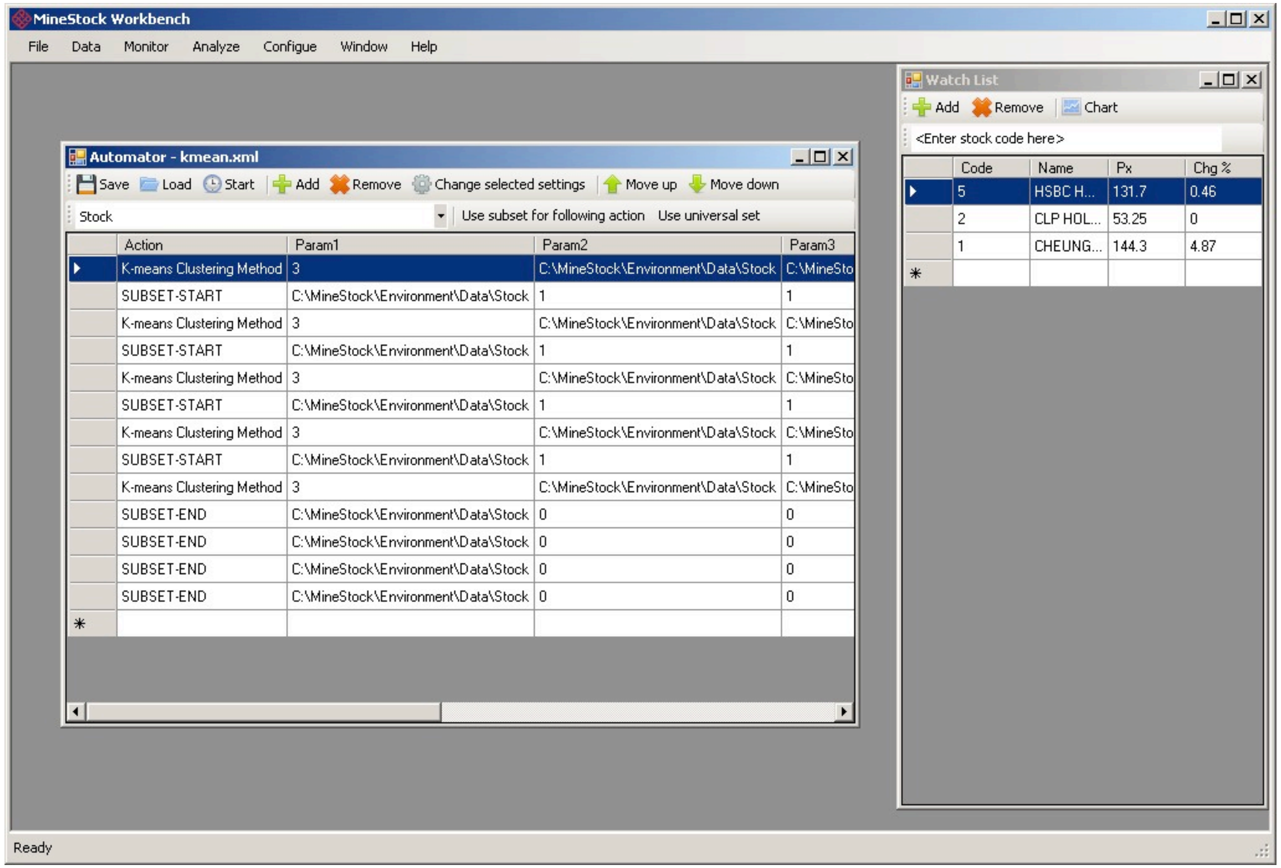
Export to

Path: C:\Temp

File:

		Last price	E(R), % p.a.	Sharpe	Beta	Alpha	%VaR, 5%	%VaR, 5% hist	$VaR, 5%	$VaR, 5% hist
		144.3	37.15	0.52	1.17	36.97	-3.23	-2.96	-4.66	-4.27
		42.85	64.54	1.17	0.93	64.4	-3.87	-3.45	-1.66	-1.48
		8.14	13.58	-0.15	0.26	13.54	-5.12	-4.69	-0.42	-0.38
		5	-29.55	-1.57	0.35	-29.6	-3.35	-3.33	-0.17	-0.17
		0.85	9.77	-0.21	0.39	9.71	-5.64	-5.29	-0.05	-0.04
		0.28	12.73	-0.1	0.64	12.64	-8.58	-6.9	-0.02	-0.02
		0.04	-80.57	-0.83	0.96	-80.71	-12.79	-11.18	-0.01	0
		1.8	169.93	0.5	1.12	169.76	-30.82	-9.47	-0.55	-0.17
		1.75	-6.3	-0.88	0.29	-6.34	-3.23	-3.1	-0.06	-0.05
		0.21	-95.56	-1.2	2.89	-95.99	-10.14	-8.4	-0.02	-0.02
		0.25	0	-0.26	-2.05	0.31	-8.43	-5.6	-0.02	-0.01
		0.38	61	0.63	-1.93	61.29	-6.61	-5.68	-0.03	-0.02
		2.35	-31.27	-0.46	0.74	-31.38	-11.95	-7.99	-0.28	-0.19
101	HANG LU...	36.2	76.62	1.36	1.32	76.42	-4.24	-4.25	-1.54	-1.54
1010	PACMOS ...	0.53	-12.91	-0.32	0.5	-12.99	-11.06	-8.84	-0.06	-0.05
1013	WAI CHU...	0.06	0	0	0	0	0	0	0	0
1019	CONVOY	0.24	-99.98	-1.09	3	-100.44	-11.52	-10.37	-0.03	-0.02

Figure 32: Exporting data in MineStock Workbench



Action	Param1	Param2	Param3
K-means Clustering Method	3	C:\MineStock\Environment\Data\Stock	C:\MineSto
SUBSET-START	C:\MineStock\Environment\Data\Stock	1	1
K-means Clustering Method	3	C:\MineStock\Environment\Data\Stock	C:\MineSto
SUBSET-START	C:\MineStock\Environment\Data\Stock	1	1
K-means Clustering Method	3	C:\MineStock\Environment\Data\Stock	C:\MineSto
SUBSET-START	C:\MineStock\Environment\Data\Stock	1	1
K-means Clustering Method	3	C:\MineStock\Environment\Data\Stock	C:\MineSto
SUBSET-START	C:\MineStock\Environment\Data\Stock	1	1
K-means Clustering Method	3	C:\MineStock\Environment\Data\Stock	C:\MineSto
SUBSET-END	C:\MineStock\Environment\Data\Stock	0	0
SUBSET-END	C:\MineStock\Environment\Data\Stock	0	0
SUBSET-END	C:\MineStock\Environment\Data\Stock	0	0
SUBSET-END	C:\MineStock\Environment\Data\Stock	0	0

Code	Name	Px	Chg %
5	HSBC H...	131.7	0.46
2	CLP HOL...	53.25	0
1	CHEUNG...	144.3	4.87

Figure 33: Defining batch actions in MineStock Workbench

action' will lead to a pop up of the dialog box shown in 34, asking for the system path of stock price and cluster information for reference. Clicking 'Use universal set' will show a similar dialog box but only requesting the input of stock path.

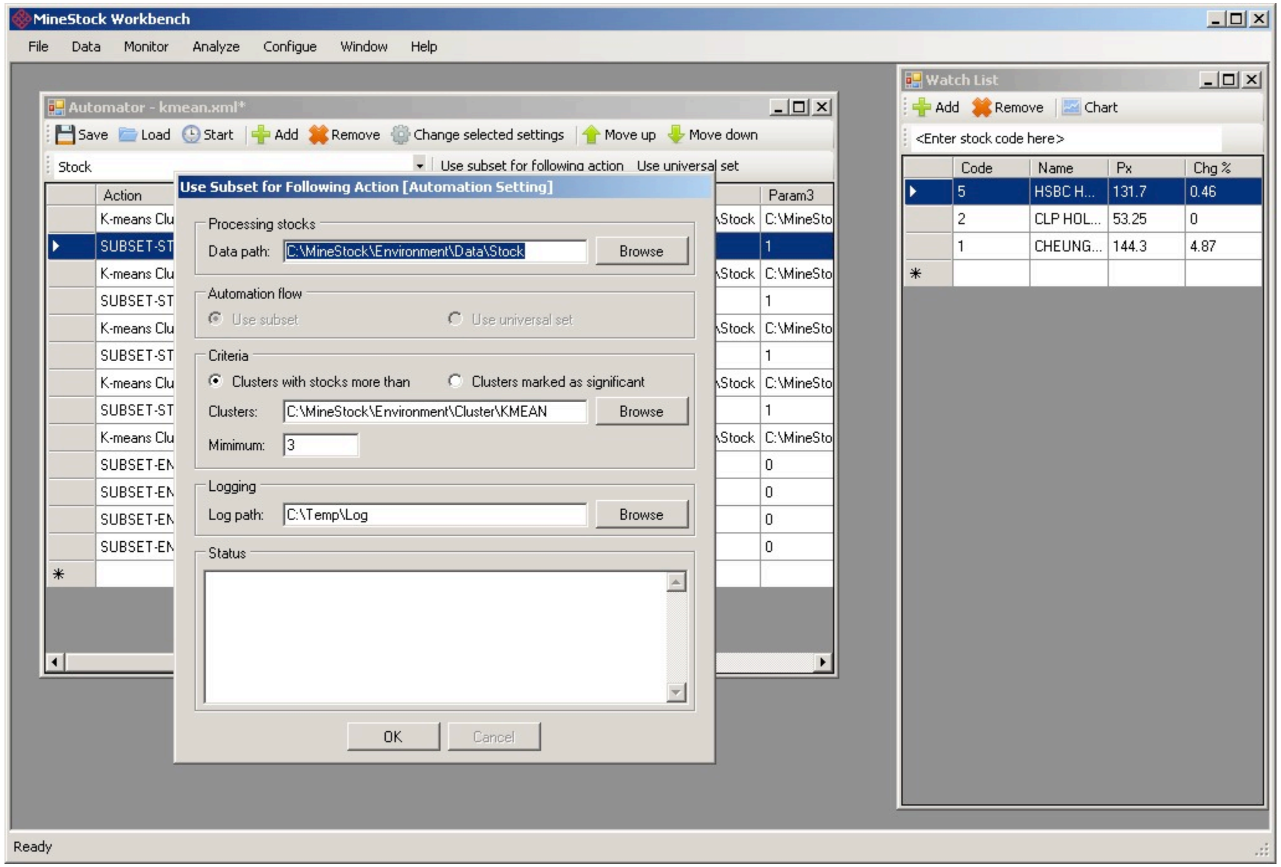
Normally the action on subset are defined in the automator in the following manner:

1. Clustering method cls
    - Number of clusters = n
    - Stock path = C:\Stock
    - Cluster path = C:\cls
2. Use subset for the following action (SUBSET-START)
    - Stock path = C:\Stock
    - Cluster path = C:\cls
    - Minimum number of stocks = m
3. Clustering method cls
    - Number of clusters = m
    - Stock path = C:\Stock
    - Cluster path = C:\cls
4. Use universal set (SUBSET-END)
    - Stock path = C:\Stock

In the above example, the clustering action in slot 3 is wrapped by the 'SUBSET-START' action in slot 2 and 'SUBSET-END' action in slot 4. Therefore, when the automator tries to run action 2, it will locate all cluster information stored in 'C:\cls', which produced by the clustering action performed in slot 1, and for each cluster identifies, see if their number of stocks inside exceeded the defined m, if the result is positive, perform clustering action in slot 3 on the stocks in that particular cluster in the 'root'. Therefore, if all of the clusters in 'root', which produced by clustering method in slot 1, has more than m stocks inside, then the clustering action in slot 3 will be executed n times. Finally, after action 4, which represents as a closure of actions on subset, everything will back to normal.

7.4.8 Execute Batch Actions

When investors defined the list automation actions using the panel, they can click the 'Start' button on the top of the automator panel as demonstrated in figure 33. After that, a dialog box same as the one in figure 35 will be shown in the system. In the dialog box, users can select if they need to keep system logs for the automation process, if so, the path of log files is required. After clicking the 'Start' button, the automation process will locate and run the actions as defined.



Action		Param3
K-means Clu	\Stock	C:\MineSto
SUBSET-ST		1
K-means Clu	\Stock	C:\MineSto
SUBSET-ST		1
K-means Clu	\Stock	C:\MineSto
SUBSET-ST		1
K-means Clu	\Stock	C:\MineSto
SUBSET-ST		1
K-means Clu	\Stock	C:\MineSto
SUBSET-EN		0
SUBSET-EN		0
SUBSET-EN		0
SUBSET-EN		0

Use Subset for Following Action [Automation Setting]

Processing stocks

Data path: C:\MineStock\Environment\Data\Stock Browse

Automation flow

● Use subset ○ Use universal set

Criteria

● Clusters with stocks more than ○ Clusters marked as significant

Clusters: C:\MineStock\Environment\Cluster\KMEAN Browse

Mimimum: 3

Logging

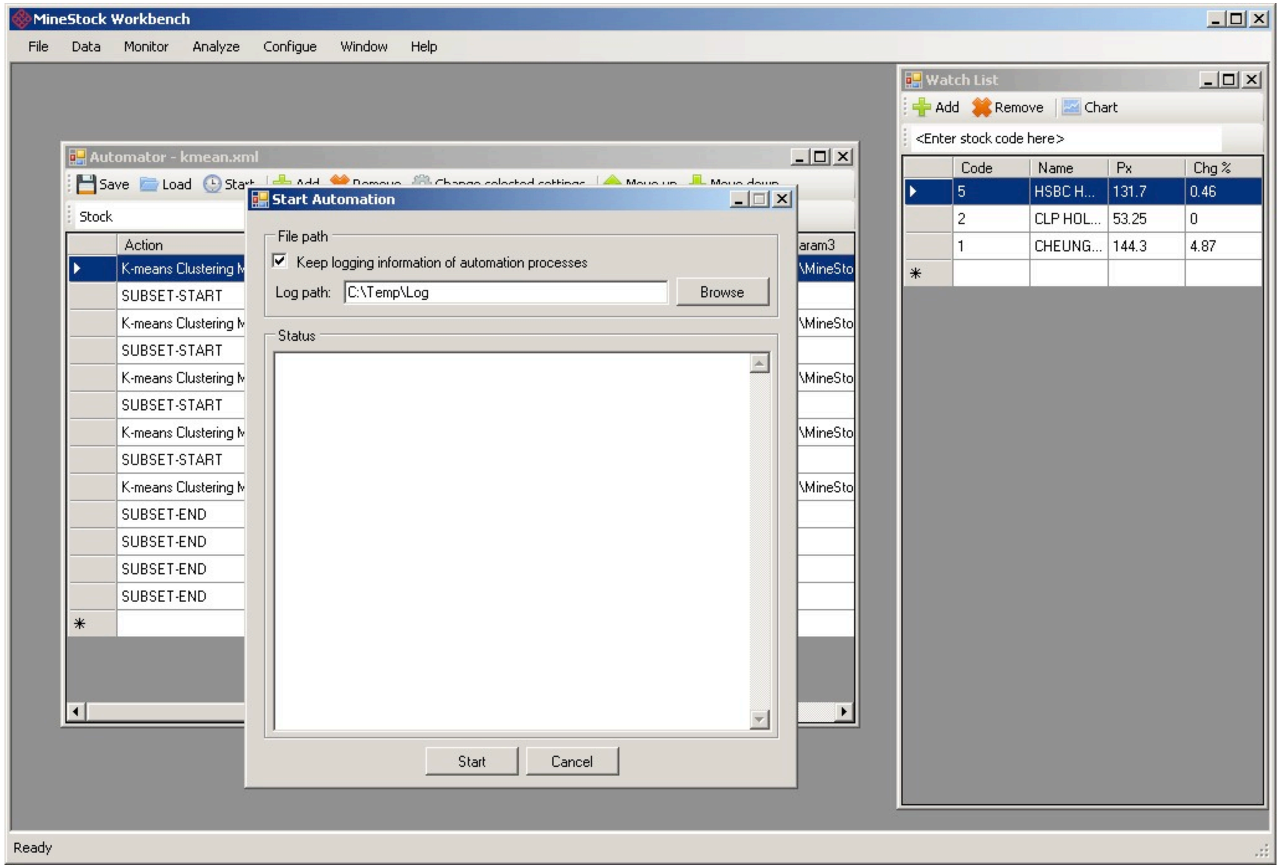
Log path: C:\Temp\Log Browse

Status

OK Cancel

Code	Name	Px	Chg %
5	HSBC H...	131.7	0.46
2	CLP HOL...	53.25	0
1	CHEUNG...	144.3	4.87

Figure 34: Defining batch actions on subset in MineStock Workbench



Stock

Action	aram3
K-means Clustering M	\MineSto
SUBSET-START	
K-means Clustering M	\MineSto
SUBSET-START	
K-means Clustering M	\MineSto
SUBSET-START	
K-means Clustering M	\MineSto
SUBSET-START	
K-means Clustering M	\MineSto
SUBSET-END	
SUBSET-END	
SUBSET-END	
SUBSET-END	
*	

File path

☑ Keep logging information of automation processes

Log path: C:\Temp\Log

Status

<Enter stock code here>

Code	Name	Px	Chg %
5	HSBC H...	131.7	0.46
2	CLP HOL...	53.25	0
1	CHEUNG...	144.3	4.87

Figure 35: Executing batch actions in MineStock Workbench

7.5 Configuring Workbench

As discussed in section 6.2, the key parts of business logic in the system are heavily modulized. The panel shown in figure 36 allows user to modify all the system paths of modules and default data path used in the system. The panel can be accessed by clicking ‘Configue’ in the menu, then ‘Environment Path’.

8 Evaluation

8.1 Grouping of Stocks

After we know the details and working steps of the algorithms, how to use them in the software tool, it is important for us to also learn about and evaluate the effectiveness of the algorithms. In this section, the writer will analyze them by looking into the grouping of stocks done by the 3 algorithms.

8.1.1 Classical K-means Technique

The followings show the result of a sample trial of k-means algorithm, processing 1302 stocks which are having 3 months of EOD price data, starting from July 2010 to September 2010. Number of clusters is defined to be 5.

Cluster #	Count of stocks
1	2
2	638
3	14
4	628
5	20
Total	1302

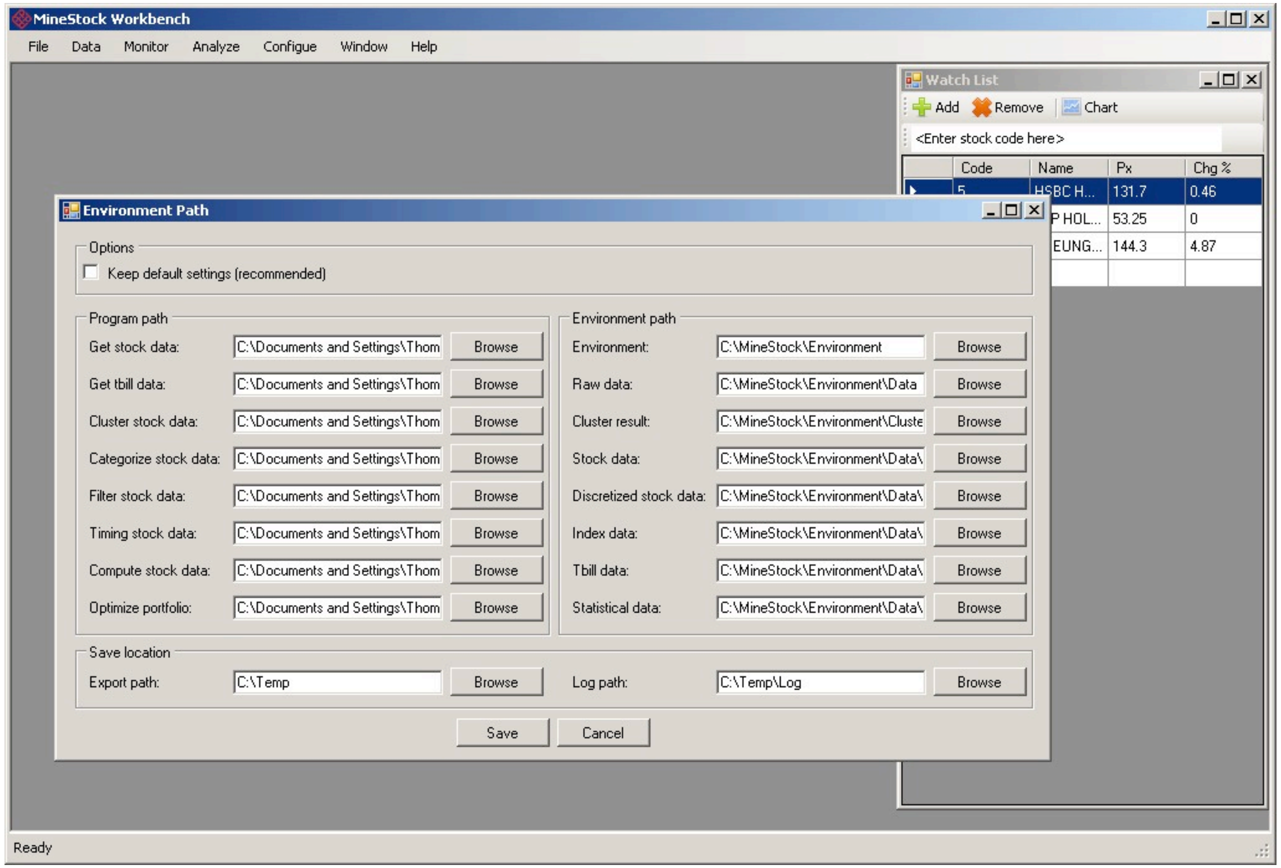
Table 2: Count of stocks in each clusters (k-means example)

We can observe that most of the stocks have gone into clusters 2 and 4. The writer’s studies on the clustering results found that most of the stocks, if they appeared to have a long term trend, either upside or downside, they will congregate into one or two clusters, resulting the other clusters contains only the outliers. For example:

Stock code	Mode daily return	Highest return	Highest return %
642.HK	+0	+0.19	+190%
8298.HK	+0	+0.15	+100%

Table 3: Detail of cluster 1 (k-means example)

The reason that most of the stocks go into one or two clusters is related to how the algorithm deal with the specialties of time series data. In the calculation of



<Enter stock code here>

	Code	Name	Px	Chg %
▶	5	HSBC H...	131.7	0.46
		P HOL...	53.25	0
		EUNG...	144.3	4.87

Options

□ Keep default settings (recommended)

Program path

Get stock data: C:\Documents and Settings\Thom

Get tbill data: C:\Documents and Settings\Thom

Cluster stock data: C:\Documents and Settings\Thom

Categorize stock data: C:\Documents and Settings\Thom

Filter stock data: C:\Documents and Settings\Thom

Timing stock data: C:\Documents and Settings\Thom

Compute stock data: C:\Documents and Settings\Thom

Optimize portfolio: C:\Documents and Settings\Thom

Environment path

Environment: C:\MineStock\Environment

Raw data: C:\MineStock\Environment\Data

Cluster result: C:\MineStock\Environment\Cluste

Stock data: C:\MineStock\Environment\Data\

Discretized stock data: C:\MineStock\Environment\Data\

Index data: C:\MineStock\Environment\Data\

Tbill data: C:\MineStock\Environment\Data\

Statistical data: C:\MineStock\Environment\Data\

Save location

Export path: C:\Temp

Log path: C:\Temp\Log

Figure 36: Configuring the MineStock Workbench

euclidean distance, the return of each day is simply treated as another attribute of the stock. Therefore, difference in distribution of short term fluctuations may end up offsetting each other. Like the case shown in the following figure:

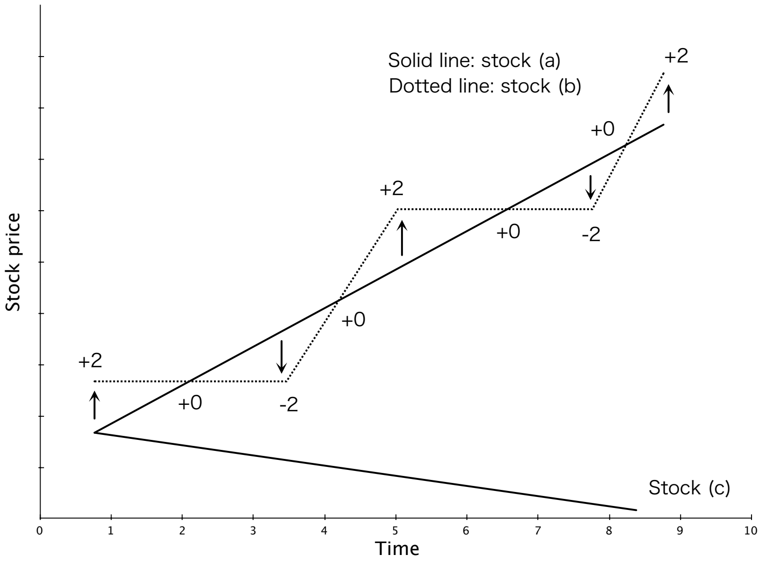


Figure 37: Only outliers are identified by k-means

In the above example, even though stock (a) and (b) have a very different short term fluctuation pattern, their euclidean distance is 0 as the fluctuations offset each other. As such, they will be assigned into the same cluster. Only stock (c), which has a very high distance with stock (a) and (b), is likely to be assigned into another separate cluster.

The studies above shows that k-means may be a effective way to locate outlier of the market. However, it is clearly not the best option to cluster the stocks evenly into groups that meets our application needs.

8.1.2 Identical Sequence Extraction

The followings show the result of a trial of this sequence extraction algorithm, processing 769 stocks with their 5 year EOD price data from the end of 2005 to end of 2010. Historical prices are discretized into 5 intervals according to their change, length of window and number of clusters is defined to be 5. A relatively balanced distribution of stocks among the 5 clusters can be observed.

Cluster #	HSI constituent stocks	Other stocks	Total Count
1	0	38	38
2	0	149	149
3	0	255	255
4	37	218	255
5	0	72	72
Total	37	732	769

Table 4: Count of stocks in each clusters (identical sequence extraction example)

Figure 38 shows the differences of selected sequence occurrences between the center of clusters, calculated with the identical settings mentioned above. It is obvious that in the result of this trial, most of the actively traded stocks as well as all the HSI constituent stocks are grouped into cluster 4. Stocks which are not actively traded and always stay in its price level, are grouped into cluster 1, 2 and 5. Stocks which have not apparent patterns are grouped into cluster 3.

8.1.3 Similar Motif Discovery

The followings show the result of a trial of this motif discovery algorithm, processing 1170 stocks which are having 1 years of EOD price data within the start of 2010 and the end of 2010. Historical prices are discretized into 5 intervals according to their change and number of clusters is defined to be 5. Similarity multiplier is set as 2. A relatively balanced distribution of stocks among the 5 clusters can be observed.

Cluster #	HSI constituent stocks	Other stocks	Total Count
1	0	205	205
2	0	65	65
3	1	314	315
4	44	419	463
5	0	122	122
Total	45	1125	1170

Table 5: Count of stocks in each clusters (similar motif discovery example)

Figure 39 shows the similarity metrics between the center of clusters and the HSI constituent stocks with the same settings. We can see that the clusters are quite distinct as they have a large distance to each other.

8.2 Effectiveness of Clustering

In this section, the writer will analyze the 3 clustering algorithms by calculating different correlation metrics. Details of steps performed to acquire the statistical figures can refer to section 5.4.

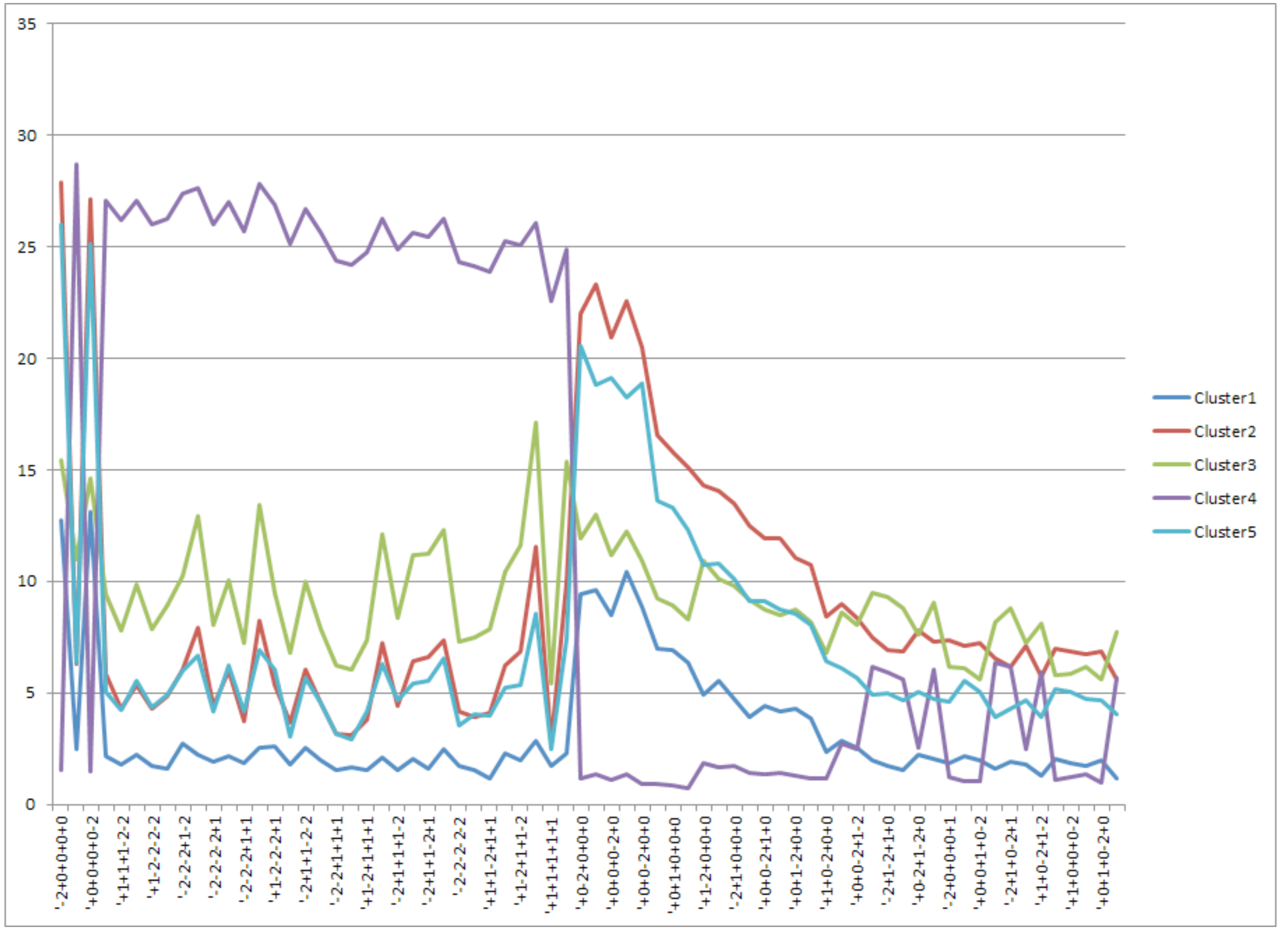


Figure 38: Difference in fluctuation between clusters (identical sequence extraction example)

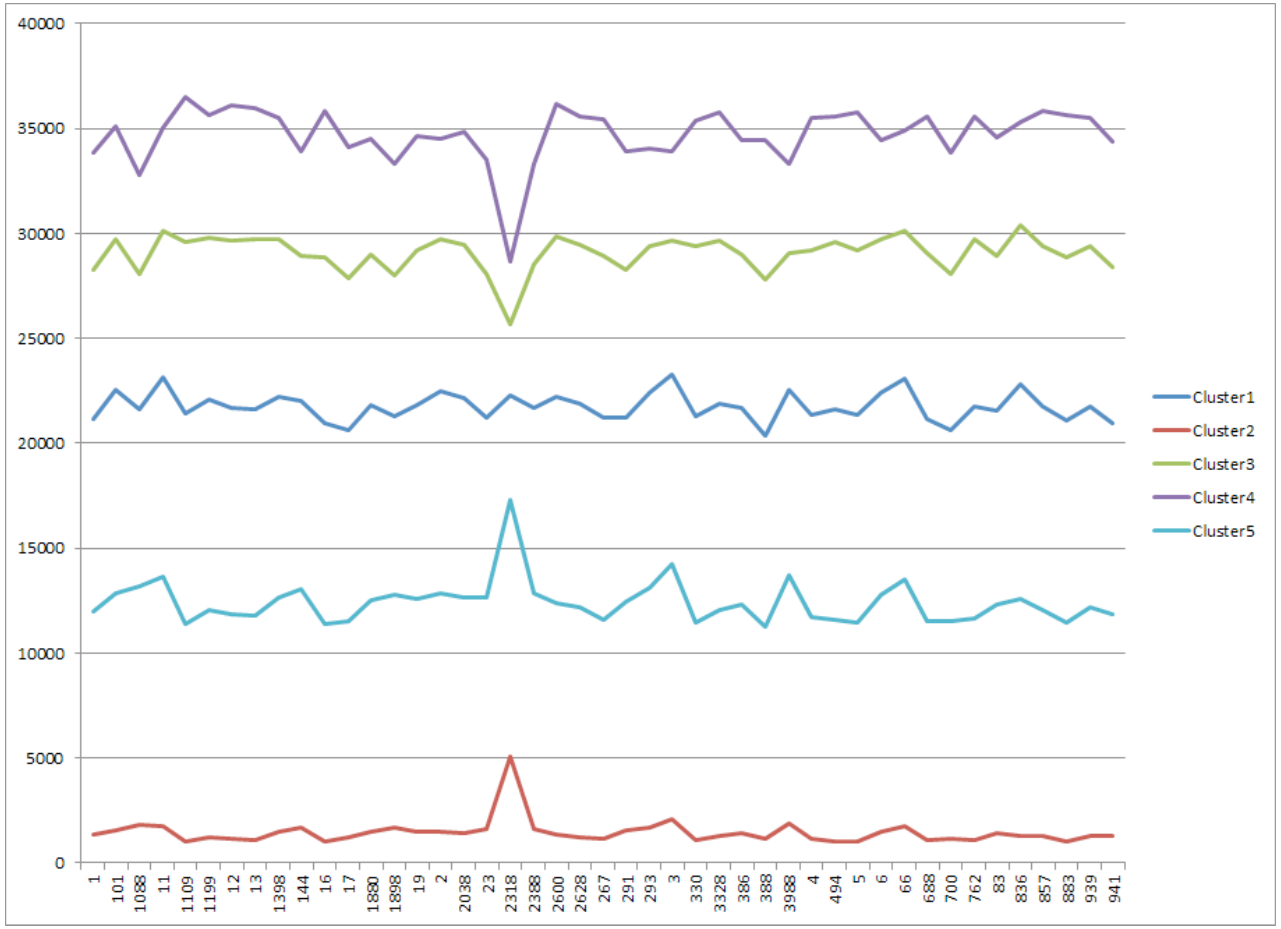


Figure 39: Difference in distance to HSI constituent stocks between clusters (similar motif discovery sample)

8.2.1 Classical K-means Technique

Below show the result of a trial of k-means algorithm, processing 815 stocks which are having 2 years of EOD price data from 2006 to 2007. Number of clusters is defined to be 3. 5 rounds of further clustering with same settings are applied on existing clusters if they have more than 3 stocks inside. Numerical breakdown for each iteration and clusters can refer to the Appendix section.

Iteration	Total clusters formed	Avg. correlation	Change
0	1 (U)	8.915%	-
1	3	9.252%	3.771%
2	5	9.547%	3.188%
3	7	10.044%	5.210%
4	11	10.492%	4.459%
5	15	10.909%	3.975%

Table 6: Correlation within clusters (k-means trial 1)

In the above table, the average correlation figure shown for iteration 0 is the one computed base on all the available stocks before any clustering process has been taken place. From the results we can see each iteration of k-means only offers a slight increase on the intra-cluster correlation of the stocks. This confirms the view stated in section 8.1.1 regarding the grouping of stocks performed by this algorithm. Since most of the clusters formed only consists of a small number of outlier stocks, the total number of clusters formed is limited. As such, users of this clustering technique not only unable to get a set of balanced clusters, in terms of number of stocks inside, by the algorithm with a high initial number of clusters parameter defined, also, even if the users try to run another iteration of clustering again on each initial clusters acquired, the benefit is not significant.

Iteration	Total clusters formed	Avg. correlation	Compare with no clusters
0	1 (U)	8.915%	-
1	3	4.091%	-54.113%
2	5	3.710%	-58.388%
3	7	5.280%	-40.774%
4	11	5.289%	-40.673%
5	15	5.675%	-36.342%

Table 7: Correlation across clusters (k-means trial 1)

From table 7, we can observe that although the stocks has a correlation of around 9% in average, the clustering process reduced the figure to 4.1% in the first iteration, and further reduced it to 3.7% in the second iteration. The bouncing back effect of the average inter-cluster correlation in later iterations is resulted by the comparison of sub-clusters, as their originating clusters has already had a relatively higher intra-cluster correlation in the later iterations.

Another trial with same settings applied except changing the number of clusters from 3 to 5 shows similar results as follows.

Iteration	Total clusters formed	Avg. correlation	Change
0	1 (U)	8.915%	-
1	5	9.602%	7.700%
2	9	10.182%	6.044%
3	13	11.327%	11.246%
4	21	13.786%	21.701%
5	37	16.428%	19.167%

Table 8: Correlation within clusters (k-means trial 2)

With the number of clusters changed from 3 to 5, as mentioned above, the average correlation within clusters are still considered undesirable although the magnitude of improvement in each further clustering iteration is higher.

Iteration	Total clusters formed	Avg. correlation	Compare with no clusters
0	1 (U)	8.915%	-
1	5	4.649%	-47.855%
2	9	4.392%	-50.736%
3	13	6.865%	-22.996%
4	21	8.735%	-2.019%
5	37	8.813%	-1.149%

Table 9: Correlation across clusters (k-means trial 2)

The results of inter-cluster correlation also show similar results in this trial. The bouncing back effect of the average correlation also observed after the second iteration. The magnitude of figures in this trial are slightly higher than the first trial. However, they still offer a lower correlation than the one calculated before clustering is performed.

8.2.2 Identical Sequence Extraction

The followings show the result of a trial of identical sequence extraction algorithm, processing 815 stocks which are having 2 years of EOD price data from 2006 to 2007. Number of clusters is defined to be 3, width of window is defined to be 5. 5 rounds of further clustering with same settings are applied on existing clusters if they have more than 3 stocks inside.

Iteration	Total clusters formed	Avg. correlation	Change
0	1 (U)	8.915%	-
1	3	10.362%	16.227%
2	9	12.797%	23.502%
3	27	16.509%	29.001%
4	72	24.053%	46.694%
5	160	36.758%	52.822%

Table 10: Correlation within clusters (identical sequence extraction trial 1)

Comparing to the intra-cluster correlation figures produced by k-means clustering algorithm, we can see that the results of identical sequence extraction algorithm shown in 10 are better. The average correlation within clusters after the second iteration is 12.8% with 9 clusters formed, 2.6% higher than trial 2 of k-means after the second iteration with same number of clusters. The improvement percentage of intra-cluster correlation figures also found to be higher and higher when further clustering iterations on existing clusters are performed. In contrast, the improvement rate of k-means have been lower down after the fifth iteration of clustering in both trial 1 and 2.

It is worth noting that the massive amount of total clusters formed may make investors even more difficult the determine which stocks are appropriate to add into their portfolios, despite having a high correlation within clusters. Therefore, users of this software tool are suggested to perform 1 or 2 times of further clustering only. The reason for doing 5 iterations with identical sequence extraction algorithm is to demonstrate the difference in performance and trend of this method in contrast with k-means. While the high correlation within sub-clusters may not be usable in our scenario, it suggests some future works which are possible to do, like prediction of price movement of stocks. Readers can refer to section 9 for more details.

Iteration	Total clusters formed	Avg. correlation	Compare with no clusters
0	1 (U)	8.915%	-
1	3	4.693%	-47.358%
2	9	7.536%	-15.469%
3	27	8.379%	-6.022%
4	72	8.563%	-3.949%
5	160	8.645%	-3.032%

Table 11: Correlation across clusters (identical sequence extraction trial 1)

From the above table of correlations across clusters, this time we see a worse result compared with k-means. It is because k-means are more effective in locating outlier, so the lower correlations between clusters of outlier with the cluster of the rest of the stocks result in a lower average inter-cluster correlations of k-means. However, it does not imply that the k-means algorithm is a better algorithm to locate stocks which are good for portfolio diversification, because the abnormal movements of the stocks within the sampled time period are not guaranteed to be sustainable. We should, instead, look for stocks which are having a low correlation due to their fundamental factors like differences in industries or market orientation.

Another trial with same settings applied except changing the number of clusters from 3 to 5 shows results as follows.

Iteration	Total clusters formed	Avg. correlation	Change
0	1 (U)	8.915%	-
1	5	11.568%	29.748%
2	24	16.251%	20.488%
3	99	29.066%	78.855%
4	290	52.258%	79.790%
5	380	54.025%	3.383%

Table 12: Correlation within clusters (identical sequence extraction trial 2)

The results of correlation within clusters remains similar after the number of clusters changed from 3 to 5. However, this time we can observe that the improvement rates of identical sequence extraction algorithm have been lower down significantly after the fifth iteration of clustering.

Iteration	Total clusters formed	Avg. correlation	Compare with no clusters
0	1 (U)	8.915%	-
1	5	6.850%	-23.165%
2	24	8.355%	-6.292%
3	99	8.608%	-3.451%
4	290	8.674%	-2.712%
5	380	8.690%	-2.532%

Table 13: Correlation across clusters (identical sequence extraction trial 2)

The results of inter-cluster correlation also show similar results in this trial. The bouncing back effect of the average correlation also observed after the first iteration. This is probably attributed by the relatively higher intra-cluster correlations observed in the first iteration compared with previous trials. The magnitude of figures in this trial are slightly higher than the first trial. However, they still offer a lower correlation than the one calculated before clustering is performed.

8.2.3 Similar Motif Discovery

The followings show the result of a trial of similar motif discovery algorithm, processing 815 stocks which are having 2 years of EOD price data from 2006 to 2007. Number of clusters is defined to be 3, similarity multiplier is defined to be 1.2. 5 rounds of further clustering with same settings are applied on existing clusters if they have more than 3 stocks inside.

Iteration	Total clusters formed	Avg. correlation	Change
0	1 (U)	8.915%	-
1	3	10.801%	21.143%
2	9	13.007%	20.426%
3	24	16.529%	27.084%
4	70	24.915%	50.733%
5	180	40.501%	62.553%

Table 14: Correlation within clusters (similar motif extraction trial 1)

Overall we can observe a similar but slightly better result compared with identical sequence extraction algorithm. This algorithm also provides slightly better improvement rate of intra-cluster correlation figures compared with identical sequence extraction algorithm. At the end of the fifth iteration, an average correlation over 40% within clusters is found in this trial.

Similar with the identical sequence extraction algorithm, the massive amount of total clusters formed by this method may make investors even more difficult the determine which stocks are appropriate to add into their portfolios, despite having a high correlation within clusters. Therefore, users of this software tool are suggested to perform 1 or 2 times of further clustering only. The reason for doing 5 iterations with identical sequence extraction algorithm is to demonstrate the difference in performance and trend of this method in contrast with k-means. While the high correlation within sub-clusters may not be usable in our scenario, it suggests some future works which are possible to do, like prediction of price movement of stocks. Readers can refer to section 9 for more details.

Iteration	Total clusters formed	Avg. correlation	Compare with no clusters
0	1 (U)	8.915%	-
1	3	6.021%	-32.471%
2	5	7.709%	-13.537%
3	7	8.420%	-5.560%
4	11	8.596%	-3.587%
5	15	8.654%	-2.934%

Table 15: Correlation across clusters (identical sequence extraction trial 1)

Similar to the case of identical sequence extraction algorithm, this time we see a worse result compared with k-means from the above table of correlations across clusters. It is because k-means are more effective in locating outlier, so the lower correlations between clusters of outlier with the cluster of the rest of the stocks result in a lower average inter-cluster correlations of k-means. However, it does not imply that the k-means algorithm is a better algorithm to locate stocks which are good for portfolio diversification, because the abnormal movements of the stocks within the sampled time period are not guaranteed to be sustainable. We should, instead, look for stocks which are having a low correlation due to their fundamental factors like differences in industries or market orientation.

Another trial with same settings applied except changing the number of clusters from 3 to 5 shows similar results as follows.

Iteration	Total clusters formed	Avg. correlation	Change
0	1 (U)	8.915%	-
1	5	11.716%	31.417%
2	24	16.557%	41.314%
3	100	29.592%	78.730%
4	290	51.749%	74.873%
5	398	56.374%	8.937%

Table 16: Correlation within clusters (similar motif extraction trial 2)

The results of correlation within clusters remains similar after the number of clusters changed from 3 to 5. However, this time we can observe that the improvement rates of identical sequence extraction algorithm have been lower down significantly after the fifth iteration of clustering.

Iteration	Total clusters formed	Avg. correlation	Compare with no clusters
0	1 (U)	8.915%	-
1	5	7.024%	-21.220%
2	24	8.342%	-6.430%
3	100	8.611%	-3.411%
4	290	8.676%	-2.685%
5	398	8.689%	-2.537%

Table 17: Correlation across clusters (identical sequence extraction trial 2)

The results of inter-cluster correlation also show similar results in this trial. The bouncing back effect of the average correlation also observed after the first iteration. This is probably attributed by the relatively higher intra-cluster correlations observed in the first iteration compared with previous trials. The magnitude of figures in this trial are slightly higher than the first trial. However, they still offer a lower correlation than the one calculated before clustering is performed.

9 Future Works Possible

Because of the time and scope constraint of this project, there is still a large room for improvement. The writer has identified some future works possible based on the output of this project, they can be mainly separated into 2 routes, further enhancing the 'MineStock Workbench', or applying the 2 clustering algorithm used here on other scenarios.

'MineStock Workbench' is currently focused on the clustering on stocks and portfolio optimization. First, other data mining techniques could also be added into the system to perform other objectives, such as stock price prediction, etc. Second, as all the clustering algorithm currently used in the system looks into the historical prices of stocks only, data mining methods on other types of data, such as textual analysis on stock news, could also be applied.

From the result of evaluation of clusters in section 8.2, we found that the identical sequence extraction algorithm and the similar motif discovery algorithm will considerably outperform the k-means method after several iterations, the high intra-cluster correlation of stocks observed may suggest that price predations done within the stock in these cluster would be very effective. Whether this hypothesis is accurate requires future investigation.

10 Conclusion

In this project, the writer targets to build a system which would allows investors to diversify their portfolio in a better way. By combining different clustering algorithm, discretization techniques with theories in mathematical finance, the tool has been done with a variety of functions and customizations.

The writer further confirmed that the identical sequence extraction algorithm and the similar motif discovery algorithm applied in the project performs better than k-means method in terms of the correlation of stocks within a cluster produced.

Several future works are possible to try in order to improve the software tool or clustering algorithm further. They include applying certain other data mining techniques on other types of data, or investigating possibilities to apply the algorithms on other scenarios.

References

[1] Hong Kong Exchanges and Clearing Limited, "Retail Investor Survey 2000," *Hong Kong Exchanges and Clearing Limited*, Mar 16, 2001. [Online]. Available: http://www.hkex.com.hk/eng/stat/research/ris/documents/oris00_e.pdf. [Accessed: Sep 22, 2010]

[2] Hong Kong Exchanges and Clearing Limited, "Retail Investor Survey 2009," *Hong Kong Exchanges and Clearing Limited*, Mar 30, 2010. [Online]. Available: http://www.hkex.com.hk/eng/stat/research/Documents/RIS2009.pdf. [Accessed: Sep 22, 2010]

[3] A. Lam, "RBS sued for sale of Lehman-linked derivatives," *South China Morning Post* (May. 13, 2010), sec. City2 col. City.

[4] Alliance of Lehman Brothers Victims, "Website of Alliance of Lehman Brothers Victims," *Alliance of Lehman Brothers Victims*, Sep 19, 2010. [Online]. Available: http://www.lbv.org.hk/. [Accessed: Sep 20, 2010]

[5] G. Gan, C. Ma, J. Wu, *Data Clustering: Theory, Algorithms, and Applications.* Philadelphia: Society for Industrial and Applied Mathematics, 2007, pp.71 & 161.

[6] J. Han, M. Kamber, *Data Mining - Concepts and Techniques*, 2nd Edition. Sao Francisco: Elsevier, 2006, pp. 493-497.

[7] C. H. Ma, C. C. Chan, "UPSEC: An Algorithm for Classifying Unaligned Protein Sequence into Functional Families." *Journal of Computational Biology*, **15**, 4, pp. 431-443, May 2008.

[8] A. Mueen, E. Keogh, Q. Zhu, S. Cash, B. Westover, "Exact Discovery of Time Series Motifs." University of California, Feb 23, 2009. [Online]. Available: http://www.siam.org/proceedings/datamining/2009/dm09_045_mueena.pdf. [Accessed: Feb 14, 2010]

[9] H. M. Markowitz, "The Early History of Portfolio Theory: 1600-1980," in *Harry Markowitz: Selected Works*, H. M. Markowitz, Ed. New Jersey: World Scientific Publishing Company, pp. 5-16, 2009.

[10] H. M. Markowitz, "Portfolio Selection," *The Journal of Finance*, **7**, 1, pp. 77-91, Mar 1952.

[11] D. McNulty, “Bettering Your Portfolio With Alpha And Beta,” *Investopedia*, Jan 11, 2011. [Online]. Available: http://www.investopedia.com/articles/07/alphabeta.asp. [Accessed: Jan 11, 2011]

[12] M. Choudhry, *An Introduction to Value-at-risk*, 4th Edition. West Sussex: Wiley, 2006, pp. 36-37.

[13] S. H. Siu, S. M. Yung, Y. C. Tang, M. K. Tong, “Value-at-risk (Simulated Loss of a Portfolio).” Course Project, Computational Finance, The Hong Kong Polytechnic University, Hong Kong, 2010.

Appendices

Table 18: Breakdown within clusters (k-means trial 1)

Iteration	Cluster	Stock count	Avg. correl.	Weighted
1	1	2	0.70241	0.00172
1	2	811	0.08962	0.08918
1	3	2	0.65723	0.00161
2	2\1	808	0.09004	0.08927
2	2\2	2	0.66626	0.00163
2	2\3	1	1.00000	0.00123
3	2\1\1	795	0.09094	0.08871
3	2\1\2	2	0.69713	0.00171
3	2\1\3	11	0.28309	0.00382

Table 19: Breakdown across clusters (k-means trial 1)

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	1	2	813	0.04205
1	2	811	4	0.04091
1	3	2	813	0.03978
2	2\1	808	7	0.03710
2	2\2	2	813	0.04512
2	2\3	1	814	0.00496
2	1	2	813	0.04205
2	3	2	813	0.03978
3	2\1\1	795	20	0.05281
3	2\1\2	2	813	0.05551
3	2\1\3	11	804	0.06175
3	2\2	2	813	0.04512
3	2\3	1	814	0.00496
3	1	2	813	0.04205
3	3	2	813	0.03978

Table 20: Breakdown within clusters (identical sequence extraction extraction trial 1)

Iteration	Cluster	Stock count	Avg. correl.	Weighted
1	1	616	0.12140	0.09176
1	2	55	0.02928	0.00198
1	3	144	0.05597	0.00989
2	1\1	257	0.20932	0.06601
2	1\2	234	0.10712	0.03076
2	1\3	125	0.07468	0.01145
2	2\1	15	0.13133	0.00242
2	2\2	20	0.10348	0.00254
2	2\3	20	0.02295	0.00056
2	3\1	44	0.08371	0.00452
2	3\2	89	0.06457	0.00705
2	3\3	11	0.19777	0.00267
3	1\1\1	95	0.21277	0.02480
3	1\1\2	69	0.32080	0.02716
3	1\1\3	93	0.17860	0.02038
3	1\2\1	54	0.12919	0.00856
3	1\2\2	106	0.12910	0.01679
3	1\2\3	74	0.11221	0.01019
3	1\3\1	59	0.08825	0.00639
3	1\3\2	45	0.10233	0.00565
3	1\3\3	21	0.15874	0.00409
3	2\1\1	11	0.17316	0.00234
3	2\1\2	1	1.00000	0.00123
3	2\1\3	3	0.51200	0.00188
3	2\2\1	7	0.25577	0.00220
3	2\2\2	1	1.00000	0.00123
3	2\2\3	12	0.16131	0.00238
3	2\3\1	1	1.00000	0.00123
3	2\3\2	16	0.00735	0.00014
3	2\3\3	3	0.50113	0.00184
3	3\1\1	11	0.19815	0.00267
3	3\1\2	18	0.14253	0.00315
3	3\1\3	15	0.16291	0.00300
3	3\2\1	53	0.07672	0.00499
3	3\2\2	17	0.15845	0.00331
3	3\2\3	19	0.14728	0.00343
3	3\3\1	5	0.37493	0.00230
3	3\3\2	2	0.68455	0.00168
3	3\3\3	4	0.42536	0.00209

Table 21: Breakdown across clusters (identical sequence extraction extraction trial 1)

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	1	616	199	0.04766
1	2	55	760	0.01736
1	3	144	671	0.05510
2	3\1	44	771	0.05323
2	3\2	89	726	0.05487
2	3\3	11	804	0.04895
2	2\1	15	800	0.01682
2	2\2	20	795	0.03124
2	2\3	20	795	0.00206
2	1\1	257	558	0.09134
2	1\2	234	581	0.08840
2	1\3	125	690	0.06864
3	3\3\1	5	810	0.05254
3	3\3\2	2	813	0.04799
3	3\3\3	4	811	0.04448
3	3\2\1	53	762	0.05237
3	3\2\2	17	798	0.05904
3	3\2\3	19	796	0.05438
3	3\1\1	11	804	0.05560
3	3\1\2	18	797	0.05006
3	3\1\3	15	800	0.05395
3	2\3\1	1	814	-0.00041
3	2\3\2	16	799	0.00047
3	2\3\3	3	812	0.01096
3	2\2\1	7	808	0.02921
3	2\2\2	1	814	0.04382
3	2\2\3	12	803	0.03091
3	2\1\1	11	804	0.01660
3	2\1\2	1	814	0.00899
3	2\1\3	3	812	0.01978
3	1\3\1	59	756	0.06808
3	1\3\2	45	770	0.06386
3	1\3\3	21	794	0.07494
3	1\2\1	54	761	0.08846
3	1\2\2	106	709	0.09574
3	1\2\3	74	741	0.08404
3	1\1\1	95	720	0.11371
3	1\1\2	69	746	0.13164
3	1\1\3	93	722	0.10979

Table 22: Breakdown within clusters (similar motif extraction trial 1)

Iteration	Cluster	Stock count	Avg. correl.	Weighted
1	1	488	0.14490	0.08676
1	2	66	0.03021	0.00245
1	3	261	0.05870	0.01880
2	1\1	155	0.12590	0.02394
2	1\2	226	0.22653	0.06282
2	1\3	107	0.10022	0.01316
2	2\1	30	0.03389	0.00125
2	2\2	34	0.07724	0.00322
2	2\3	2	0.69898	0.00172
2	3\1	88	0.08026	0.00867
2	3\2	63	0.07636	0.00590
2	3\3	110	0.06959	0.00939
3	1\1\1	30	0.16239	0.00598
3	1\1\2	55	0.13470	0.00909
3	1\1\3	70	0.15274	0.01312
3	1\2\1	82	0.19522	0.01964
3	1\2\2	89	0.22984	0.02510
3	1\2\3	55	0.35400	0.02389
3	1\3\1	15	0.17918	0.00330
3	1\3\2	68	0.11454	0.00956
3	1\3\3	24	0.15659	0.00461
3	2\1\1	4	0.39716	0.00195
3	2\1\2	15	0.00000	0.00000
3	2\1\3	11	0.17080	0.00231
3	2\2\1	14	0.15723	0.00270
3	2\2\2	6	0.31436	0.00231
3	2\2\3	14	0.15502	0.00266
3	3\1\1	13	0.17583	0.00280
3	3\1\2	5	0.35450	0.00217
3	3\1\3	70	0.09289	0.00798
3	3\2\1	28	0.11913	0.00409
3	3\2\2	14	0.16316	0.00280
3	3\2\3	21	0.13618	0.00351
3	3\3\1	36	0.09946	0.00439
3	3\3\2	35	0.11468	0.00492
3	3\3\3	39	0.09784	0.00468

Table 23: Breakdown across clusters (similar motif extraction trial 1)

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	1	488	327	0.06238
1	2	66	749	0.02281

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	3	261	554	0.06560
2	3\1	88	727	0.07064
2	3\2	63	752	0.05487
2	3\3	110	705	0.05924
2	2\1	30	785	0.00847
2	2\2	34	781	0.03290
2	2\3	2	813	0.03723
2	1\1	155	660	0.09489
2	1\2	226	589	0.09594
2	1\3	107	708	0.08221
3	3\3\1	36	779	0.05494
3	3\3\2	35	780	0.06080
3	3\3\3	39	776	0.05994
3	3\2\1	28	787	0.05578
3	3\2\2	14	801	0.04718
3	3\2\3	21	794	0.05729
3	3\1\1	13	802	0.05132
3	3\1\2	5	810	0.05039
3	3\1\3	70	745	0.07487
3	2\2\1	14	801	0.03947
3	2\2\2	6	809	0.01474
3	2\2\3	14	801	0.03322
3	2\1\1	4	811	0.01011
3	2\1\2	15	800	0.00000
3	2\1\3	11	804	0.01896
3	1\3\1	15	800	0.07170
3	1\3\2	68	747	0.08424
3	1\3\3	24	791	0.08283
3	1\2\1	82	733	0.11322
3	1\2\2	89	726	0.11714
3	1\2\3	55	760	0.13778
3	1\1\1	30	785	0.09361
3	1\1\2	55	760	0.09266
3	1\1\3	70	745	0.10257
3	2\3	2	813	0.03723

Table 24: Breakdown within clusters (k-means trial 2)

Iteration	Cluster	Stock count	Avg. correl.	Weighted
1	1	2	0.75275	0.00185
1	2	2	0.67032	0.00164
1	3	2	0.68463	0.00168
1	4	807	0.08998	0.08910
1	5	2	0.71230	0.00175
2	4\1	801	0.09071	0.08915
2	4\2	1	1.00000	0.00123

Iteration	Cluster	Stock count	Avg. correl.	Weighted
2	4\3	1	1.00000	0.00123
2	4\4	2	0.66676	0.00164
2	4\5	2	0.67560	0.00166
3	4\1\1	1	1.00000	0.00123
3	4\1\2	1	1.00000	0.00123
3	4\1\3	644	0.10018	0.07916
3	4\1\4	1	1.00000	0.00123
3	4\1\5	154	0.09400	0.01776

Table 25: Breakdown across clusters (k-means trial 2)

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	1	2	813	0.05645
1	2	2	813	0.00907
1	3	2	813	0.06008
1	4	807	8	0.04649
1	5	2	813	0.05932
2	4\1	801	14	0.04393
2	4\2	1	814	0.05868
2	4\3	1	814	0.05663
2	4\4	2	813	0.02313
2	4\5	2	813	0.03916
2	1	2	813	0.05645
2	2	2	813	0.00907
2	3	2	813	0.06008
2	5	2	813	0.05932
3	4\1\1	1	814	0.04296
3	4\1\2	1	814	0.04897
3	4\1\3	644	171	0.06880
3	4\1\4	1	814	0.11237
3	4\1\5	154	661	0.07034
3	4\2	1	814	0.05868
3	4\3	1	814	0.05663
3	4\4	2	813	0.02313
3	4\5	2	813	0.03916
3	1	2	813	0.05645
3	2	2	813	0.00907
3	3	2	813	0.06008
3	5	2	813	0.05932

Table 26: Breakdown within clusters (identical sequence extraction trial 2)

Iteration	Cluster	Stock Count	Avg. Correl	Weighted
1	1	251	0.07677	0.02364

Iteration	Cluster	Stock Count	Avg. Correl	Weighted
1	2	40	0.06487	0.00318
1	3	376	0.17131	0.07903
1	4	24	0.02930	0.00086
1	5	124	0.05884	0.00895
2	1\1	99	0.10973	0.01333
2	1\2	49	0.09246	0.00556
2	1\3	69	0.08613	0.00729
2	1\4	25	0.14815	0.00454
2	1\5	9	0.24013	0.00265
2	2\1	6	0.30749	0.00226
2	2\2	10	0.21994	0.00270
2	2\3	12	0.15854	0.00233
2	2\4	1	1.00000	0.00123
2	2\5	11	0.17176	0.00232
2	3\1	99	0.21706	0.02637
2	3\2	88	0.29318	0.03166
2	3\3	64	0.14080	0.01106
2	3\4	62	0.14612	0.01112
2	3\5	63	0.17448	0.01349
2	4\1	15	0.00000	0.00000
2	4\2	2	0.66641	0.00164
2	4\3	3	0.51200	0.00188
2	4\4	2	0.67001	0.00164
2	4\5	2	0.66613	0.00163
2	5\1	31	0.10414	0.00396
2	5\2	24	0.11429	0.00337
2	5\3	9	0.21782	0.00241
2	5\4	29	0.11460	0.00408
2	5\5	31	0.10518	0.00400
3	1\1\1	32	0.15003	0.00589
3	1\1\2	14	0.20328	0.00349
3	1\1\3	25	0.15942	0.00489
3	1\1\4	9	0.23503	0.00260
3	1\1\5	19	0.21956	0.00512
3	1\2\1	9	0.26435	0.00292
3	1\2\2	14	0.17056	0.00293
3	1\2\3	5	0.37866	0.00232
3	1\2\4	11	0.20996	0.00283
3	1\2\5	10	0.22211	0.00273
3	1\3\1	15	0.19063	0.00351
3	1\3\2	11	0.21724	0.00293
3	1\3\3	21	0.14286	0.00368
3	1\3\4	13	0.18686	0.00298
3	1\3\5	9	0.22039	0.00243
3	1\4\1	2	0.72290	0.00177
3	1\4\2	7	0.32169	0.00276
3	1\4\3	11	0.22922	0.00309

Iteration	Cluster	Stock Count	Avg. Correl	Weighted
3	1\4\4	2	0.67036	0.00165
3	1\4\5	3	0.56169	0.00207
3	1\5\1	3	0.51372	0.00189
3	1\5\2	1	1.00000	0.00123
3	1\5\3	1	1.00000	0.00123
3	1\5\4	2	0.67289	0.00165
3	1\5\5	2	0.66457	0.00163
3	2\1\1	1	1.00000	0.00123
3	2\1\2	1	1.00000	0.00123
3	2\1\3	1	1.00000	0.00123
3	2\1\4	1	1.00000	0.00123
3	2\1\5	2	0.67812	0.00166
3	2\2\1	4	0.43862	0.00215
3	2\2\2	2	0.69435	0.00170
3	2\2\3	1	1.00000	0.00123
3	2\2\4	1	1.00000	0.00123
3	2\2\5	2	0.69048	0.00169
3	2\3\1	2	0.67266	0.00165
3	2\3\2	2	0.67948	0.00167
3	2\3\3	2	0.66475	0.00163
3	2\3\4	4	0.40002	0.00196
3	2\3\5	2	0.62542	0.00153
3	2\5\1	1	1.00000	0.00123
3	2\5\2	3	0.50422	0.00186
3	2\5\3	2	0.64331	0.00158
3	2\5\4	1	1.00000	0.00123
3	2\5\5	4	0.40623	0.00199
3	3\1\1	7	0.39342	0.00338
3	3\1\2	33	0.29222	0.01183
3	3\1\3	8	0.35011	0.00344
3	3\1\4	31	0.22519	0.00857
3	3\1\5	20	0.27871	0.00684
3	3\2\1	19	0.39537	0.00922
3	3\2\2	11	0.37539	0.00507
3	3\2\3	28	0.30434	0.01046
3	3\2\4	3	0.64469	0.00237
3	3\2\5	27	0.34348	0.01138
3	3\3\1	4	0.43992	0.00216
3	3\3\2	28	0.17337	0.00596
3	3\3\3	8	0.36047	0.00354
3	3\3\4	11	0.26443	0.00357
3	3\3\5	13	0.24647	0.00393
3	3\4\1	18	0.21559	0.00476
3	3\4\2	23	0.18860	0.00532
3	3\4\3	12	0.24539	0.00361
3	3\4\4	4	0.51131	0.00251
3	3\4\5	5	0.39631	0.00243

Iteration	Cluster	Stock Count	Avg. Correl	Weighted
3	3\5\1	2	0.72316	0.00177
3	3\5\2	34	0.19412	0.00810
3	3\5\3	13	0.29195	0.00466
3	3\5\4	6	0.35271	0.00260
3	3\5\5	8	0.36946	0.00363
3	5\1\1	2	0.69487	0.00171
3	5\1\2	5	0.34726	0.00213
3	5\1\3	6	0.31756	0.00234
3	5\1\4	12	0.19814	0.00292
3	5\1\5	6	0.29834	0.00220
3	5\2\1	5	0.38541	0.00236
3	5\2\2	6	0.31749	0.00234
3	5\2\3	3	0.52757	0.00194
3	5\2\4	2	0.68151	0.00167
3	5\2\5	8	0.24401	0.00240
3	5\3\1	1	1.00000	0.00123
3	5\3\2	1	1.00000	0.00123
3	5\3\3	2	0.70674	0.00173
3	5\3\4	3	0.52594	0.00194
3	5\3\5	2	0.66622	0.00163
3	5\4\1	3	0.63877	0.00235
3	5\4\2	12	0.18632	0.00274
3	5\4\3	3	0.53408	0.00197
3	5\4\4	6	0.32320	0.00238
3	5\4\5	5	0.39589	0.00243
3	5\5\1	1	1.00000	0.00123
3	5\5\2	8	0.25467	0.00250
3	5\5\3	1	1.00000	0.00123
3	5\5\4	11	0.20731	0.00280
3	5\5\5	10	0.23787	0.00292

Table 27: Breakdown across clusters (identical sequence extraction trial 2)

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	1	251	564	0.07644
1	2	40	775	0.03162
1	3	376	439	0.07538
1	4	24	791	0.00466
1	5	124	691	0.05582
2	1\1	99	716	0.08571
2	1\2	49	766	0.05803
2	1\3	69	746	0.06931
2	1\4	25	790	0.07982
2	1\5	9	806	0.05977
2	2\1	6	809	0.01771

Iteration	Cluster	Stock Count	Comparison Count	Correlation
2	2\2	10	805	0.04849
2	2\3	12	803	0.02765
2	2\4	1	814	0.03004
2	2\5	11	804	0.02635
2	3\1	99	716	0.11437
2	3\2	88	727	0.12466
2	3\3	64	751	0.09683
2	3\4	62	753	0.09826
2	3\5	63	752	0.10760
2	4\1	15	800	0.00000
2	4\2	2	813	0.00163
2	4\3	3	812	0.01978
2	4\4	2	813	0.01851
2	4\5	2	813	0.00430
2	5\1	31	784	0.05553
2	5\2	24	791	0.04745
2	5\3	9	806	0.04400
2	5\4	29	786	0.05878
2	5\5	31	784	0.05719
3	1\1\1	32	783	0.08699
3	1\1\2	14	801	0.08198
3	1\1\3	25	790	0.08617
3	1\1\4	9	806	0.06003
3	1\1\5	19	796	0.10024
3	1\2\1	9	806	0.06889
3	1\2\2	14	801	0.05222
3	1\2\3	5	810	0.05675
3	1\2\4	11	804	0.06048
3	1\2\5	10	805	0.05359
3	1\3\1	15	800	0.07630
3	1\3\2	11	804	0.06977
3	1\3\3	21	794	0.06876
3	1\3\4	13	802	0.07011
3	1\3\5	9	806	0.05227
3	1\4\1	2	813	0.09030
3	1\4\2	7	808	0.08298
3	1\4\3	11	804	0.07471
3	1\4\4	2	813	0.06430
3	1\4\5	3	812	0.09357
3	1\5\1	3	812	0.05057
3	1\5\2	1	814	0.06375
3	1\5\3	1	814	0.06710
3	1\5\4	2	813	0.06669
3	1\5\5	2	813	0.06079
3	2\1\1	1	814	0.02757
3	2\1\2	1	814	-0.02862
3	2\1\3	1	814	0.02095

Iteration	Cluster	Stock Count	Comparison Count	Correlation
3	2\1\4	1	814	0.02720
3	2\1\5	2	813	0.02980
3	2\2\1	4	811	0.04372
3	2\2\2	2	813	0.04419
3	2\2\3	1	814	0.05620
3	2\2\4	1	814	0.06996
3	2\2\5	2	813	0.04739
3	2\3\1	2	813	0.02600
3	2\3\2	2	813	0.04895
3	2\3\3	2	813	0.03297
3	2\3\4	4	811	0.01932
3	2\3\5	2	813	0.01796
3	2\5\1	1	814	0.05303
3	2\5\2	3	812	0.02557
3	2\5\3	2	813	0.03082
3	2\5\4	1	814	0.01236
3	2\5\5	4	811	0.02097
3	3\1\1	7	808	0.11621
3	3\1\2	33	782	0.13143
3	3\1\3	8	807	0.11859
3	3\1\4	31	784	0.11233
3	3\1\5	20	795	0.12551
3	3\2\1	19	796	0.14656
3	3\2\2	11	804	0.12774
3	3\2\3	28	787	0.13075
3	3\2\4	3	812	0.13607
3	3\2\5	27	788	0.14092
3	3\3\1	4	811	0.06512
3	3\3\2	28	787	0.09591
3	3\3\3	8	807	0.11730
3	3\3\4	11	804	0.09773
3	3\3\5	13	802	0.09960
3	3\4\1	18	797	0.10187
3	3\4\2	23	792	0.09912
3	3\4\3	12	803	0.09133
3	3\4\4	4	811	0.12216
3	3\4\5	5	810	0.09298
3	3\5\1	2	813	0.12368
3	3\5\2	34	781	0.10836
3	3\5\3	13	802	0.10932
3	3\5\4	6	809	0.09456
3	3\5\5	8	807	0.12298
3	5\1\1	2	813	0.07573
3	5\1\2	5	810	0.04469
3	5\1\3	6	809	0.06330
3	5\1\4	12	803	0.05960
3	5\1\5	6	809	0.04035

Iteration	Cluster	Stock Count	Comparison Count	Correlation
3	5\2\1	5	810	0.05278
3	5\2\2	6	809	0.05289
3	5\2\3	3	812	0.04870
3	5\2\4	2	813	0.05349
3	5\2\5	8	807	0.03724
3	5\3\1	1	814	0.01690
3	5\3\2	1	814	0.01465
3	5\3\3	2	813	0.06586
3	5\3\4	3	812	0.05996
3	5\3\5	2	813	0.02545
3	5\4\1	3	812	0.03258
3	5\4\2	12	803	0.05470
3	5\4\3	3	812	0.06500
3	5\4\4	6	809	0.06336
3	5\4\5	5	810	0.07370
3	5\5\1	1	814	0.04874
3	5\5\2	8	807	0.05285
3	5\5\3	1	814	-0.01399
3	5\5\4	11	804	0.05873
3	5\5\5	10	805	0.06577
3	2\4	1	814	0.03004
3	4\1	15	800	0.00000
3	4\2	2	813	0.00163
3	4\3	3	812	0.01978
3	4\4	2	813	0.01851
3	4\5	2	813	0.00430

Table 28: Breakdown within clusters (similar motif extraction trial 2)

Iteration	Cluster	Stock count	Avg. correl.	Weighted
1	\1	37	0.03321	0.00151
1	\2	350	0.17773	0.07633
1	\3	55	0.06529	0.00441
1	\4	224	0.08568	0.02355
1	\5	149	0.06223	0.01138
2	1\1	4	0.42685	0.00209
2	1\2	7	0.25059	0.00215
2	1\3	4	0.39716	0.00195
2	1\4	7	0.27018	0.00232
2	1\5	15	0.00000	0.00000
2	2\1	40	0.13895	0.00682
2	2\2	91	0.14938	0.01668
2	2\3	104	0.29036	0.03705
2	2\4	66	0.20046	0.01623
2	2\5	49	0.23817	0.01432

Iteration	Cluster	Stock count	Avg. correl.	Weighted
2	3\1	9	0.21476	0.00237
2	3\2	3	0.51152	0.00188
2	3\3	13	0.19200	0.00306
2	3\4	14	0.15655	0.00269
2	3\5	16	0.14954	0.00294
2	4\1	63	0.10047	0.00777
2	4\2	21	0.14524	0.00374
2	4\3	37	0.13829	0.00628
2	4\4	37	0.10810	0.00491
2	4\5	66	0.12784	0.01035
2	5\1	33	0.09206	0.00373
2	5\2	32	0.10298	0.00404
2	5\3	27	0.12347	0.00409
2	5\4	4	0.43246	0.00212
2	5\5	53	0.09190	0.00598
3	1\2\1	2	0.66139	0.00162
3	1\2\2	2	0.67948	0.00167
3	1\2\3	1	1.00000	0.00123
3	1\2\4	1	1.00000	0.00123
3	1\2\5	1	1.00000	0.00123
3	1\4\1	1	1.00000	0.00123
3	1\4\2	2	0.66710	0.00164
3	1\4\3	1	1.00000	0.00123
3	1\4\4	1	1.00000	0.00123
3	1\4\5	2	0.67755	0.00166
3	2\1\1	16	0.20615	0.00405
3	2\1\2	15	0.22672	0.00417
3	2\1\3	3	0.53326	0.00196
3	2\1\4	2	0.58783	0.00144
3	2\1\5	4	0.44431	0.00218
3	2\2\1	17	0.21156	0.00441
3	2\2\2	11	0.27128	0.00366
3	2\2\3	24	0.20472	0.00603
3	2\2\4	6	0.37645	0.00277
3	2\2\5	33	0.19191	0.00777
3	2\3\1	23	0.42918	0.01211
3	2\3\2	18	0.44363	0.00980
3	2\3\3	3	0.65714	0.00242
3	2\3\4	34	0.23344	0.00974
3	2\3\5	26	0.33963	0.01083
3	2\4\1	10	0.32569	0.00400
3	2\4\2	8	0.33659	0.00330
3	2\4\3	31	0.23102	0.00879
3	2\4\4	6	0.39743	0.00293
3	2\4\5	11	0.34207	0.00462
3	2\5\1	7	0.36803	0.00316
3	2\5\2	14	0.32857	0.00564

Iteration	Cluster	Stock count	Avg. correl.	Weighted
3	2\5\3	8	0.37771	0.00371
3	2\5\4	3	0.57797	0.00213
3	2\5\5	17	0.32514	0.00678
3	3\1\1	1	1.00000	0.00123
3	3\1\2	3	0.48304	0.00178
3	3\1\3	1	1.00000	0.00123
3	3\1\4	2	0.69926	0.00172
3	3\1\5	2	0.66725	0.00164
3	3\3\1	3	0.52981	0.00195
3	3\3\2	2	0.67219	0.00165
3	3\3\3	4	0.42542	0.00209
3	3\3\4	3	0.53981	0.00199
3	3\3\5	1	1.00000	0.00123
3	3\4\1	2	0.67437	0.00165
3	3\4\2	4	0.40187	0.00197
3	3\4\3	1	1.00000	0.00123
3	3\4\4	6	0.31514	0.00232
3	3\4\5	1	1.00000	0.00123
3	3\5\1	3	0.52409	0.00193
3	3\5\2	4	0.39804	0.00195
3	3\5\3	2	0.66854	0.00164
3	3\5\4	3	0.50228	0.00185
3	3\5\5	4	0.43295	0.00212
3	4\1\1	25	0.13703	0.00420
3	4\1\2	5	0.38515	0.00236
3	4\1\3	17	0.19254	0.00402
3	4\1\4	3	0.53935	0.00199
3	4\1\5	13	0.20828	0.00332
3	4\2\1	2	0.66343	0.00163
3	4\2\2	6	0.35136	0.00259
3	4\2\3	5	0.34559	0.00212
3	4\2\4	6	0.33942	0.00250
3	4\2\5	2	0.67275	0.00165
3	4\3\1	3	0.53599	0.00197
3	4\3\2	9	0.26834	0.00296
3	4\3\3	10	0.26983	0.00331
3	4\3\4	10	0.28090	0.00345
3	4\3\5	5	0.37226	0.00228
3	4\4\1	3	0.49881	0.00184
3	4\4\2	3	0.53491	0.00197
3	4\4\3	7	0.27522	0.00236
3	4\4\4	13	0.21372	0.00341
3	4\4\5	11	0.21385	0.00289
3	4\5\1	15	0.23726	0.00437
3	4\5\2	14	0.21975	0.00377
3	4\5\3	13	0.22694	0.00362
3	4\5\4	15	0.21547	0.00397

Iteration	Cluster	Stock count	Avg. correl.	Weighted
3	4\5\5	9	0.28210	0.00312
3	5\1\1	6	0.30584	0.00225
3	5\1\2	5	0.36382	0.00223
3	5\1\3	11	0.19855	0.00268
3	5\1\4	4	0.41195	0.00202
3	5\1\5	7	0.27470	0.00236
3	5\2\1	6	0.33176	0.00244
3	5\2\2	8	0.26591	0.00261
3	5\2\3	3	0.50474	0.00186
3	5\2\4	8	0.23461	0.00230
3	5\2\5	7	0.30332	0.00261
3	5\3\1	3	0.48813	0.00180
3	5\3\2	12	0.21839	0.00322
3	5\3\3	7	0.27799	0.00239
3	5\3\4	2	0.67166	0.00165
3	5\3\5	3	0.53346	0.00196
3	5\5\1	1	1.00000	0.00123
3	5\5\2	23	0.13889	0.00392
3	5\5\3	10	0.24633	0.00302
3	5\5\4	16	0.15167	0.00298
3	5\5\5	3	0.54320	0.00200

Table 29: Breakdown across clusters (similar motif extraction trial 2)

Iteration	Cluster	Stock Count	Comparison Count	Correlation
1	1	37	778	0.00997
1	2	350	465	0.07876
1	3	55	760	0.04449
1	4	224	591	0.08056
1	5	149	666	0.05917
2	1\1	4	811	0.01348
2	1\2	7	808	0.02191
2	1\3	4	811	0.01011
2	1\4	7	808	0.01547
2	1\5	15	800	0.00000
2	2\1	40	775	0.08790
2	2\2	91	724	0.10284
2	2\3	104	711	0.12092
2	2\4	66	749	0.11386
2	2\5	49	766	0.11892
2	3\1	9	806	0.02987
2	3\2	3	812	0.03923
2	3\3	13	802	0.06384
2	3\4	14	801	0.04196
2	3\5	16	799	0.03743

Iteration	Cluster	Stock Count	Comparison Count	Correlation
2	4\1	63	752	0.07677
2	4\2	21	794	0.06695
2	4\3	37	778	0.08594
2	4\4	37	778	0.06747
2	4\5	66	749	0.08961
2	5\1	33	782	0.04781
2	5\2	32	783	0.05350
2	5\3	27	788	0.06780
2	5\4	4	811	0.05928
2	5\5	53	762	0.06131
3	1\2\1	2	813	0.00265
3	1\2\2	2	813	0.04895
3	1\2\3	1	814	0.01356
3	1\2\4	1	814	0.00918
3	1\2\5	1	814	0.02639
3	1\4\1	1	814	0.00598
3	1\4\2	2	813	0.00455
3	1\4\3	1	814	0.01812
3	1\4\4	1	814	0.02720
3	1\4\5	2	813	0.02425
3	2\1\1	16	799	0.09035
3	2\1\2	15	800	0.09724
3	2\1\3	3	812	0.05724
3	2\1\4	2	813	0.05569
3	2\1\5	4	811	0.08372
3	2\2\1	17	798	0.09873
3	2\2\2	11	804	0.10423
3	2\2\3	24	791	0.10654
3	2\2\4	6	809	0.11090
3	2\2\5	33	782	0.10654
3	2\3\1	23	792	0.15115
3	2\3\2	18	797	0.15188
3	2\3\3	3	812	0.15949
3	2\3\4	34	781	0.11508
3	2\3\5	26	789	0.13759
3	2\4\1	10	805	0.11943
3	2\4\2	8	807	0.11300
3	2\4\3	31	784	0.11482
3	2\4\4	6	809	0.10855
3	2\4\5	11	804	0.13063
3	2\5\1	7	808	0.11355
3	2\5\2	14	801	0.12310
3	2\5\3	8	807	0.12350
3	2\5\4	3	812	0.09358
3	2\5\5	17	798	0.13089
3	3\1\1	1	814	-0.00585
3	3\1\2	3	812	0.03005

Iteration	Cluster	Stock Count	Comparison Count	Correlation
3	3\1\3	1	814	0.02978
3	3\1\4	2	813	0.05260
3	3\1\5	2	813	0.02448
3	3\3\1	3	812	0.06766
3	3\3\2	2	813	0.05107
3	3\3\3	4	811	0.07532
3	3\3\4	3	812	0.05750
3	3\3\5	1	814	0.05023
3	3\4\1	2	813	0.02799
3	3\4\2	4	811	0.02940
3	3\4\3	1	814	0.07892
3	3\4\4	6	809	0.05308
3	3\4\5	1	814	0.01389
3	3\5\1	3	812	0.04161
3	3\5\2	4	811	0.02041
3	3\5\3	2	813	0.03588
3	3\5\4	3	812	0.05821
3	3\5\5	4	811	0.03655
3	4\1\1	25	790	0.07271
3	4\1\2	5	810	0.06762
3	4\1\3	17	798	0.08558
3	4\1\4	3	812	0.04786
3	4\1\5	13	802	0.08158
3	4\2\1	2	813	0.04038
3	4\2\2	6	809	0.07776
3	4\2\3	5	810	0.05394
3	4\2\4	6	809	0.07919
3	4\2\5	2	813	0.05538
3	4\3\1	3	812	0.06129
3	4\3\2	9	806	0.08186
3	4\3\3	10	805	0.08841
3	4\3\4	10	805	0.10022
3	4\3\5	5	810	0.07517
3	4\4\1	3	812	0.05245
3	4\4\2	3	812	0.06483
3	4\4\3	7	808	0.05310
3	4\4\4	13	802	0.08184
3	4\4\5	11	804	0.06330
3	4\5\1	15	800	0.10372
3	4\5\2	14	801	0.09123
3	4\5\3	13	802	0.08405
3	4\5\4	15	800	0.08611
3	4\5\5	9	806	0.08216
3	5\1\1	6	809	0.03515
3	5\1\2	5	810	0.05215
3	5\1\3	11	804	0.05351
3	5\1\4	4	811	0.05330

Iteration	Cluster	Stock Count	Comparison Count	Correlation
3	5\1\5	7	808	0.04163
3	5\2\1	6	809	0.05750
3	5\2\2	8	807	0.04523
3	5\2\3	3	812	0.04159
3	5\2\4	8	807	0.04651
3	5\2\5	7	808	0.07134
3	5\3\1	3	812	0.06243
3	5\3\2	12	803	0.07241
3	5\3\3	7	808	0.05404
3	5\3\4	2	813	0.08508
3	5\3\5	3	812	0.07234
3	5\5\1	1	814	0.09341
3	5\5\2	23	792	0.06144
3	5\5\3	10	805	0.07855
3	5\5\4	16	799	0.04851
3	5\5\5	3	812	0.05661
3	1\1	4	811	0.01348
3	1\3	4	811	0.01011
3	1\5	15	800	0.00000
3	3\2	3	812	0.03923
3	5\4	4	811	0.05928