

國立清華大學
統計學研究所

碩 士 論 文

利用 RankNet 排序結果推論突變發生之先後關係

**Inference of Mutation Order through the
RankNet Algorithm**

所 別 統計學研究所

組 別 生物統計組

指導教授 謝文萍 博士 (Wen-Ping Hsieh, Ph.D.)

姓 名 賴汶靖 (Wen-Ching Lai)

學 號 104024511

中 華 民 國 一 百 零 六 年 七 月

題目

利用 RankNet 排序結果推論突變發生之先後關係

摘要

突變會造成等位基因頻率(allele frequency)的改變，等位基因頻率越高代表突變發生的時間越早。RankNet 是利用機器學習的方式，來解決排序上的問題，將兩兩錯排的機率降到最低。我們依照等位基因頻率的高低，利用 RankNet 的演算法，找出突變發生的先後順序。由於單一樣本不足以支持不同基因突變之間的因果關係，我們希望藉由多個樣本來推測這些重複發生的演化關係。我們提出的方法先利用近鄰連結法(Neighbor-joining method)來建構一顆無根的樹，再從這棵樹中找出使排序錯誤最低的樹根，排序的結論是利用 RankNet 計算得到。這篇研究的目標是利用不同樣本之間的等位基因頻率，來推測突變發生的先後順序及突變的因果關係，藉此幫助我們探索癌細胞的進化過程。

Title

Inference of Mutation Order through the RankNet Algorithm

Abstract

Allele frequency is the relative frequency of a variant at a particular locus, and larger cellular occupancy of a mutation is associated with earlier mutations. RankNet is a machine learning algorithm that aim to reduce pairwise ranking errors. We use it to infer the right order of mutation occurrence. Since the evolutionary order of gene mutations by only one sample is not informative to any causal relationship, we use multiple samples to infer the causal relationship of those mutations. We proposed a procedure to solve this problem. Our first step is to use Neighbor joining method to constructed an unrooted tree according to the allele frequencies of mutations. We then find the optimal tree by choosing a root with minimum rank error rate. The aim of this thesis is to use the mutation allele frequencies from multiple samples to infer the the right order of mutation occurrence and reconstruct the most likely phylogenetic tree for the recurrent mutations of cancer cells.

CONTENT

1. Introduction	1
2. Materials and Methods	4
2.1 Materials	4
2.2 Two-way Poisson Mixture Model	4
2.3 RankNet	6
2.4 Neighbor-joining Method	12
2.5 Allocate the Root of the Tree	17
3. Results	20
3.1 Simulation	20
3.2 On real dataset	27
4. Conclusion and Discussion	31
Reference	34
Appendix	36



1. Introduction

Cancer is a dynamic disease that develops over time. Various types of somatic changes to chromosome structure lead to gain or loss of sections of DNA. Since tumor heterogeneity brings about an ongoing challenge in the field of personalized treatment, it has been a worth noting problem that affects clinical strategies. To better understand the progression of this disease, phylogenetic tree reconstruction that infers the ancestral relationship between somatic mutations has been an important problem.

The hypothesis that all cells within a tumor are originated from a single founder cell has been proposed in the 1970s (Nowell, P.C.) [1]. This unicellular origin concept also states that tumor cells with selective growth advantage become more competitive than normal cells and thus permit cells to expand. The single founder cell is a diploid tumor cell with growth advantage, and cancer progression is resulted from acquiring genetic variability within the original clone. With the subsequent appearance of more favorable mutations that cause clonal expansion over time, a tumor consisting of multiple subpopulations of cells is thus formed.

In order to construct a phylogenetic tree that contains the evolutionary relationships among those subpopulations of mutated genes, single nucleotide variants (SNV) and copy number aberrations (CNA) are the two types of data that is widely used. Tai et al. proposed a two-way Poisson mixture model that only requires read depth information and solely focus on CNA to infer clonal evolution structures [2]. The model provides the copy number and mutational cellular prevalence (MCP) for each locus. Although the relative sizes of the mutational cellular prevalence provide some hints about the

evolutionary order of the mutations, Tai's method did not explicitly infer the relationship. In the current study, we will reconstruct the most frequent evolutionary sequences among the copy number mutations with the output provided by the two-way Poisson mixture model.

Most of the sequencing projects collect multiple random samples of the same disease. The two-way Poisson mixture model estimates the cellular occupancy of each copy number mutations within each sample. Larger cellular occupancy implies earlier mutations. The information is used to calculate the mutational distance of any pair of mutations across samples. The evolutionary tree of mutations is constructed with the neighbor-joining tree. Since it is an unrooted tree, we need to find the most likely rooted tree to interpret the evolutionary sequences. We first use RankNet to sort the order of presence of gene mutations according to their cellular proportions within each sample. Next, a root is selected to minimize pairwise rank error.

The procedure will be evaluated through the simulation data and then applied on a set of Head and Neck cancer data downloaded from TCGA (The Cancer Genome Atlas) database. Cancer pathways and gene mutation functional groups are also important issues for exploring cancer progression. It is believed that there exist rules that govern the transformation of normal cells into malignant cancers. Identifying these cancer pathways and their interconnections will be crucial for the development of effective targeted therapies [3]. The probable pathways can be inferred through the upstream and downstream of that particular mutation on the tree.

The thesis is arranged as follows. Section 2 described our methods, including the two-way Poisson mixture model, the RankNet algorithm, neighbor-joining method

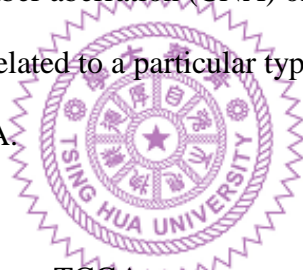
and tree construction. Section 3 showed both the results for the simulated data and for a Head and Neck cancer dataset from TCGA. Section 4 is the conclusion for this thesis and some discussion for future work or improvements.



2. Materials and Methods

2.1 Materials

The real data we use for analysis is a set of head and neck cancer data from The Cancer Genome Atlas (TCGA). The data consists of whole exon sequencing information of 75 pairs of tumor/control samples. The set includes 20,846 genes with 180,243 exons. The exon level data is quite messy and we only choose the genes with clear copy number change for the complete analysis. Genes with consistent amplification or deletion states in more than 25% of the exons for any sample were retained as background gene set. A total of 3,244 background genes were selected. Here, we discuss the copy number aberration (CNA) on the genes. Thus, every gene in the background gene set is related to a particular type of mutation, mutation A would stand for CNA of gene A.



Since the clinical records for those TCGA are not complete, the association analysis is applied only on 68 paired samples. Among the 68 patients, 32 developed either invasion or metastasis and they seem to have acquired more heterogeneity. We only consider this subset for the phylogenetic tree reconstruction.

2.2 Two-way Poisson Mixture Model

Tumor tissue structure can be decomposed into two concepts, subclonal cellular prevalence (SCP) and mutational cellular prevalence (MCP). SCP is the fraction of homogenous cells carrying the same set of mutations, and MCP is the fraction of cells carrying a certain type of mutation (Figure 1). Two-way Poisson mixture model assumes that the read depth of each locus follows a Poisson distribution with a mean

proportional to a function of MCP and its copy number.

SCPs carrying the same set of mutations can be added up to match a particular gene's MCP according to the evolution structure of subclones as in Figure 1. The data we use throughout the thesis would be in the form of MCPs, which is the fraction of cells carrying that mutation. The value would be a number in $[0, 1]$, and it is used to infer the mutation time. Values closer to 1 means the mutation was acquired earlier in time, and values closer to 0 means the mutation was acquired more recently.

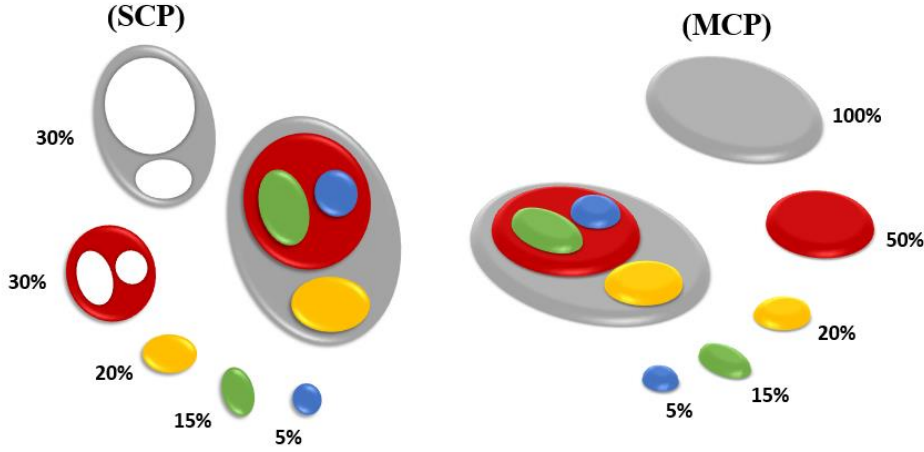


Figure 1. An illustration for SCP and MCP.

Suppose all the loci considered can be grouped into m_1 types of copy number states and m_2 different sizes of MCPs. Let X_i be the read depth of locus- i . The two-way Poisson mixture model is formulated as

$$P(X_i | \tilde{r}, a_{base,i}) = \sum_{k=1}^{m_1} \sum_{h=1}^{m_2} \pi_{kh} f_{kh}(X_i), \quad \forall i \quad (1)$$

, where each $f_{kh}(X_i)$ is the sampling distribution of the read depth for the k th group of copy number state, and the h th group of MCPs. $a_{base,i}$ is an normalization constant estimated by half of the average read depth at region- i . π_{kh} are the mixture

weights. $f(x)$ is specified as a Poisson distribution with mean $\mu_{kh} = a_{base,i} \times (2(1 - r_h) + c_k r_h)$. Hence, the maximum likelihood estimator of π_{kh} and \tilde{r} are estimated via E-M algorithm.

2.3 RankNet

Although the two-way Poisson mixture model provides the cellular proportions of the mutations within each sample, and the relative sizes of mutations within each sample can be compared, we do not have enough evidence to infer the evolutionary order between the mutations unless the relationship holds across multiple samples. We hence need a strategy to integrate the pairwise information within each sample into a global ranking conclusion from multiple samples.

Learning to Rank (LTR) is a class of machine learning techniques that is used to solve ranking problems. It is a central part of many information retrieval problems, such as search engine ranking, collaborative filtering and recommendation system. Ranking models can be classified into three approaches according to their input and the loss function they try to minimize: pointwise approach, pairwise approach and listwise approach [4].

Pairwise algorithms are trained for ranking the order of documents according to the information of pairwise comparison. Documents are items with different features and what we are interested is the ranking for documents. For a given collection of documents, a relative score system tells which document should be ranked higher. In our study, genes can be regarded as documents, with MCPs of different samples taken as different features. The features are presented by some quantified numbers. The goal for this system is to reduce cases where the pairs of results are in the wrong order, and

to minimize the number of inversions in ranking. Genes with higher MCP values are expected to be ranked higher.

Suppose that X is the input space (feature space) consisting of lists of feature vectors, and Y is the output space consisting of lists of grades. For a pair of instance $\{A, B\}$, if A is to be ranked higher than B , we write it as $A \triangleright B$. Assume that $f(\cdot)$ is a function mapping from $x \in X$ to $y \in Y$. Then the goal of the learning task is to automatically come up with a well learned function $f(\cdot)$ such that when instance $i \triangleright j$, the corresponding grades $y_i = f(x_i)$ is larger than $y_j = f(x_j)$.

Note that in pairwise ranking problems, no particular assumption for transitivity is needed. The numerical values of the grades of the documents are not important since those ranking methods does not care much about the exact score that each document gets. They only care more about the relative ordering among all instances [5].

RankNet is a pairwise ranking method based on neural network model. It builds the relationship with a probabilistic model, and only target probabilities that reflect the relationship of each training pairs are needed. For each training pair, if the rank of x_i is higher than x_j , the target probability \bar{P}_{ij} is defined as 1. If the ranks are the same, the target probability \bar{P}_{ij} is defined as $\frac{1}{2}$. When the rank is of the reverse order, the target probability \bar{P}_{ij} is defined as 0. We keep the notation \bar{P}_{ij} for the underlying truth following the original paper [6]. Hence, the input for this model is $\{(x_i, x_j), \bar{P}_{ij}\}$. We consider ranking models $f: X \rightarrow Y$ such that $f(x_1) > f(x_2)$ is taken to assert that $x_1 \triangleright x_2$.

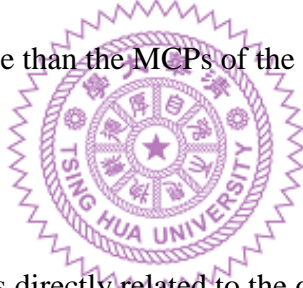
For each iteration, the algorithm approximates \bar{P}_{ij} with P_{ij} , which is called the modeled posterior in [6] and stands for $P(x_i \succ x_j)$. P_{ij} is generated according to the mapping function $f(\cdot)$. RankNet takes the following cross entropy loss function as cost.

$$C_{ij} \equiv C(y_{ij}) = -\bar{P}_{ij} \cdot \log(P_{ij}) - (1 - \bar{P}_{ij}) \cdot \log(1 - P_{ij}) \quad (2)$$

Loss function is used to evaluate the prediction result of our mapping function $f(\cdot)$.

If the feature vectors are scored correctly, then the loss will be small.

Here P_{ij} is the output probabilities in each iteration, and we determine the target probability \bar{P}_{ij} by $\bar{P}_{ij} = 1$ if the MCP of the i th gene is larger than the MCP of the j th gene in more than half of the samples considered. When only half of the samples get larger MCPs of the i th gene than the MCPs of the j th gene, $\bar{P}_{ij} = \frac{1}{2}$. Otherwise, $\bar{P}_{ij} = 0$.



The probability measure P_{ij} is directly related to the output, y_{ij} , of the mapping function $f(\cdot)$ and it has to take values between 0 and 1. Hence, we use the logistic function to map the output values y_{ij} to $[0, 1]$ as follows.

$$P_{ij} = \frac{\exp(y_{ij})}{1 + \exp(y_{ij})} = \frac{1}{1 + \exp(-y_{ij})} \quad (3)$$

, where $y_i = f(x_i)$, $y_j = f(x_j)$ and $y_{ij} \equiv f(x_i) - f(x_j)$.

The loss function can be transformed into

$$\begin{aligned} C_{ij} &= -\bar{P}_{ij} \cdot \left(\log\left(\frac{1}{1 + e^{-y_{ij}}}\right) - \log\left(\frac{e^{-y_{ij}}}{1 + e^{-y_{ij}}}\right) \right) - \log\left(\frac{e^{-y_{ij}}}{1 + e^{-y_{ij}}}\right) \\ &= -\bar{P}_{ij} \cdot \log\left(\frac{1}{e^{-y_{ij}}}\right) + \log\left(\frac{e^{-y_{ij}} + 1}{e^{-y_{ij}}}\right) \\ &= -\bar{P}_{ij} \cdot y_{ij} + \log(1 + e^{y_{ij}}) \end{aligned} \quad (4)$$

Then, the cost is in the form of mapping function output, y_{ij} , and the target probability \bar{P}_{ij} .

The advantage of RankNet using cross entropy loss function is that it is more robust with noisy labels. C_{ij} is comfortably symmetric with minimum value equal to $\log(2)$ at $y_{ij} = 0$. That means the loss function gives a cost that does not equal to zero, for the model when it cannot tell whether instance i or instance j should be ranked on top.

RankNet uses neural network models as the framework (Figure 2). It is a net with a single output node, and it adopts our cross entropy loss function as cost. One single output node means the net outputs one ranking score for each instance. If the score is higher, then it has a higher ranking, which means that it should be place on the top of the ranking list. The model iteratively train parameters of each layer to minimize the cross entropy loss function stated above.

Let the activation function of each node in the j th layer be $g^{(j)}$. An activation function defines the output of a node given an input, and the role of the function in a neural network is to produce a non-linear decision boundary via linear combinations of the weighted inputs. Sigmoid functions such as hyperbolic tangent and logistic function is often used. We used hyperbolic tangent as activation function in our analysis, that is $g(x) = \frac{1+e^{-2x}}{1+e^{-2x}}$. If α_k are the parameters of the model, then a gradient descent step updates the parameters by $\delta\alpha_k = -\eta_k \cdot \frac{\partial C}{\partial \alpha_k}$ in each iteration, where η_k are positive learning rates pre-specified with small numbers.

For a net with one hidden layer consisting of L nodes, an input feature vector $x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \in X \subseteq R^d$, $i = 1, 2, \dots, m$, has output $y_i = f(x_i)$ of the form

$$y_i = g^{(2)} \left(\sum_{l=1}^L w_{l1}^{(2)} \cdot g^{(1)} \left(\sum_{k=1}^d w_{kl}^{(1)} \cdot x_{ik} + b_k^{(1)} \right) + b_1^{(2)} \right) \quad (5)$$

, where w are the weights with the subscript meaning which nodes in the layers it joined and the superscript meaning the layer it was in. Hence, $w_{kl}^{(1)}$ is the weight from node k to node l in the input layer to the hidden layer, and $w_{l1}^{(2)}$ is the weight from node l in the hidden layer to the node on the output layer. $b_k^{(1)}$ denotes the offset value for the k th node in the hidden layer, and $b_1^{(2)}$ is the offset for the node on the output layer.

That is, the features of x_i multiplied by some weights $w^{(1)}$ plus an offset $b^{(1)}$ are sent into the hidden layer and they are summed up as the input of the activation function $g^{(1)}$. The temporarily output is then, again multiplied by some weights $w^{(2)}$ plus $b^{(2)}$ and they are summed up and transformed by function $g^{(2)}$ to get the model output y_i . (Figure 2)

In each iteration, a pair of instances with feature vectors and target probability

$\{(x_i, x_j), \bar{P}_{ij}\} \in X \times X \times \{0, \frac{1}{2}, 1\}$ is randomly chosen from the training set. A forward propagation is performed for x_i , then a forward propagation is performed for x_j which computes y_i and y_j , respectively. From the model output in equation (5) and the fixed target, $y_{ij} = y_i - y_j$ and \bar{P}_{ij} are plugged into our cross entropy loss function to compute the loss C_{ij} in equation (4). It then use gradient descent to update the model parameters and minimize the total loss as equations (6)-(9).

$$\frac{\partial C_{ij}}{\partial b_1^{(2)}} = C'_{ij}(g_i^{(2)'} - g_j^{(2)'}) \equiv \Delta_i^{(2)} - \Delta_j^{(2)} \quad (6)$$

$$\frac{\partial C_{ij}}{\partial w_{n1}^{(2)}} = \Delta_i^{(2)} \cdot g_{in}^{(1)} - \Delta_j^{(2)} \cdot g_{jn}^{(1)} \quad (7)$$

$$\frac{\partial C_{ij}}{\partial b_n^{(1)}} = \Delta_i^{(2)} \cdot w_{n1}^{(2)} \cdot g_{in}^{(1)'} - \Delta_j^{(2)} \cdot w_{n1}^{(2)} \cdot g_{jn}^{(1)'} \equiv \Delta_{in}^{(1)} - \Delta_{jn}^{(1)} \quad (8)$$

$$\frac{\partial C_{ij}}{\partial w_{mn}^{(1)}} = \Delta_{in}^{(1)} \cdot x_{im} - \Delta_{jn}^{(1)} \cdot x_{jm} \quad (9)$$

, where C'_{ij} is the partial derivative of C_{ij} , which is a function related to the derivative of the activation functions g and the weights w in the net. We use the apostrophe symbol to denote the derivative of a function and use the delta symbol to denote the product of C' and g' . The subscripts correspond to parameter labels and the superscripts correspond to the layers of the net.

The algorithm repeatedly iterates and updates the parameters until it reduces the total loss to an acceptable low value that would not further change much. Since the above terms all take the form of difference depending on x_i and x_j , RankNet is accomplished by a straightforward modification of back propagation neural network model [7]. The algorithm is shown in the appendix.

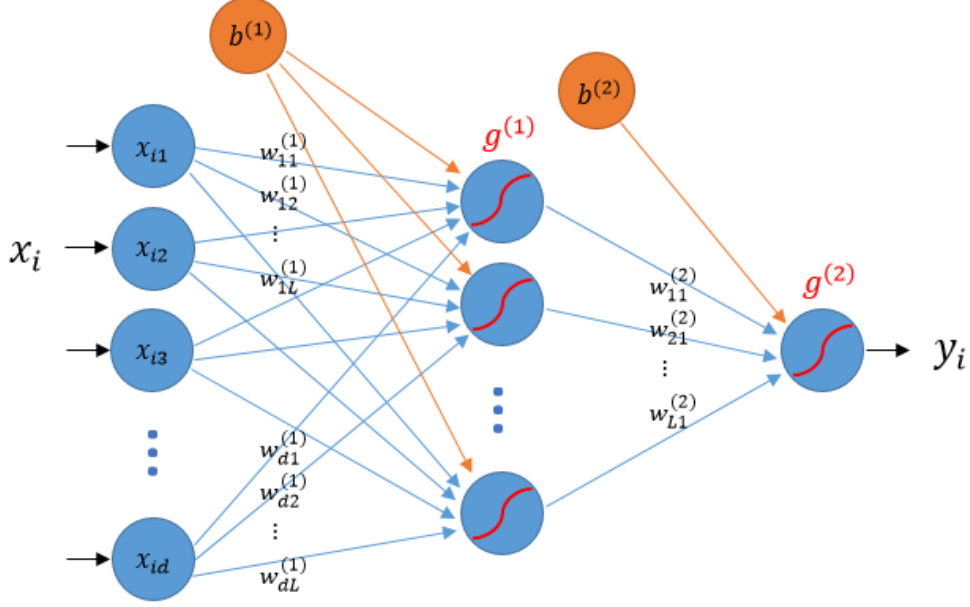


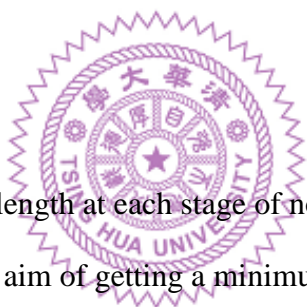
Figure 2. A schematic diagram of RankNet with one hidden layer consisting L nodes

In our case with the tumor evolution analysis, mutations are considered as the documents x_i 's with feature values from each sample referring to MCP values. In each iteration, a pair of mutations is picked and put into the model. The model then updates until optimal weights are gained and then it will output a score that refers to the rankings of each mutation. Mutation with higher scores are ranked on top, meaning that the MCPs for that gene mutation across all samples are relatively higher, and the mutation for the gene occur earlier in time.

2.4 Neighbor-joining Method

Phylogenetic tree construction methods are used to decide which mutations have the most similar behaviors across multiple samples and which mutations should be joined together on the evolution tree. Unweighted Pair Group Method with Arithmetic Mean (UPGMA)[8] and Neighbor-joining (NJ)[9] method are both frequently used methods for phylogenetic tree construction based on distance matrices. UPGMA provides rooted tree as a result, while NJ tree is unrooted [10]. It is because UPGMA is

hierarchical clustering with average linkage that adopts the ultrametric assumption. Ultrametricity assumption is called the molecular clock since it assumes a constant evolutionary rate and equal evolution time so the distances from the root to every branch tip are equal. This molecular clock assumption might not be true for the tumor evolution and it does not allow unequal distances from the branch point to the leaf node. NJ has the advantage that it does not assume all lineages evolve at the same rate or evolve for the same time period. The branch length can represent either the time lapse between gene mutations under a constant evolutionary rate or the difference of mutation accumulation rate for the same time period. In either situation, the MCP values are proportional to the branch length. We use MCPs across multiple samples to compute the distance matrix for NJ. We used NJ method for construction of our evolutionary tree.



NJ minimizes the total branch length at each stage of node merging. It provides a unique unrooted tree under the aim of getting a minimum evolution tree. Nodes connected through a single interior node are called “neighbors”. NJ method starts off with a star-like tree (Figure 3A), then sequentially join two nodes as neighbors so that the conjunction minimize total branch length, which means these two nodes has to be closely related to each other [9].

NJ method not only joins the closest points but also takes all other points into consideration. That is, at each iteration, two points that are furthest from the rest will be joined together. Since our data is the MCPs of each mutation of genes of multiple samples, and we try to infer the relationships for these mutations, we use NJ method to construct our unrooted tree. The tree first joins mutations of similar size of MCPs across samples.

We briefly summarize the algorithm as follows. Let D_{ij} be the distance between leaves i and j . In our MCP data, D_{ij} was calculated as the Euclidean distance of the MCPs between mutations i and j across samples. L_{XY} is defined as the branch lengths between node X and node Y . S_{ij} is the sum of branch lengths when i and j are joined as neighbors. Total branch length is what we try to minimize in each iteration. $S_0 = \sum_{i=1}^N L_{iX} = \frac{1}{N-1} \sum_{1 \leq i < j \leq N} D_{ij}$ is the total branch length of the star-like tree in the beginning such as the example in Figure 3A. N is the number of leave nodes, that is, the number of mutations we have in the dataset. X is initial joint node of all leaves.

Take nodes 1 and 2 for example, as in Figure 3B

$$S_{12} = (L_{1X} + L_{2X}) + L_{XY} + \sum_{i=3}^N L_{iY} \quad (10)$$

$$L_{XY} = \frac{1}{2(N-2)} \left[\sum_{k=3}^N (D_{1k} + D_{2k}) - (N-2)(L_{1X} + L_{2X}) - 2 \sum_{i=3}^N L_{iY} \right] \quad (11)$$

Since we only have D_{ij} 's as input, $L_{1X} + L_{2X} = D_{12}$ and $\sum_{i=3}^N L_{iY} =$

$\frac{1}{N-3} \sum_{3 \leq i < j \leq N} D_{ij}$ can be plugged into the above equations and get

$$\begin{aligned} S_{12} &= L_{XY} + (L_{1X} + L_{2X}) + \sum_{i=3}^N L_{iY} \\ &= \frac{1}{2(N-2)} \sum_{k=3}^N (D_{1k} + D_{2k}) + \frac{1}{2} D_{12} + \frac{1}{N-2} \sum_{3 \leq i < j \leq N} D_{ij} \end{aligned} \quad (12)$$

In each iteration, join nodes i and j with the smallest S_{ij} . Then, define the distance between node X and the rest of the nodes, and enter these distances into a new distance matrix. In our example of Figure 3, node 1 and 2 are the first to be joined at node X . Hence, we define the distance of X to all the nodes other than node 1 and node 2 as

$$D_{Xj} = \frac{1}{2}(D_{1j} + D_{2j}) \quad (13)$$

The distances computed in equation (13) are used as the distances D_{ij} in the new distance matrix to decide the next branch-out point.

Next, define the distance of X to node 1 and node 2 as

$$L_{1X} = \frac{1}{2}(D_{12} + D_{1Z} - D_{2Z}) \quad (14)$$

$$L_{2X} = \frac{1}{2}(D_{12} + D_{2Z} - D_{1Z}) \quad (15)$$

, where $D_{1Z} = \frac{1}{N-2}(\sum_{k=3}^N D_{1k})$ and $D_{2Z} = \frac{1}{N-2}(\sum_{k=3}^N D_{2k})$. The distances computed in equations (14)-(15) are the branch length of the tree for the final output and they are not used in any further iterations of the tree construction. If the average distance from node 1 to all the other nodes is longer than the average distance from node 2 to all the other nodes, the distance from node 1 to their branching point X is longer than the distance from node 2 to X .

NJ method constructs an unrooted additive tree, and the distance between the leaves measured on the tree is a good reflection of distance specified in the distance matrix D_{ij} . The method was proposed by Saitou and Nei in 1987. A year later, a correction was sent by Studier and Keppler to make the computation more efficient [11]. They

define $R_i = \sum_{k=1}^N D_{ik}$ and $M_{ij} = D_{ij} - \frac{R_i + R_j}{2}$.

Since

$$\sum_{3 \leq i < j \leq N} D_{ij} = \sum_{1 \leq i < j \leq N} D_{ij} - \sum_{1 \leq i \leq N} (D_{1i} + D_{2i}) + D_{12} \quad (16)$$

We can plug equation (16) into equation (12) and get

$$\begin{aligned} S_{12} &= \frac{1}{2(N-2)} \sum_{k=3}^N (D_{1k} + D_{2k}) + \frac{1}{2} D_{12} + \frac{1}{(N-2)} \sum_{3 \leq i < j \leq N} D_{ij} \\ &= \frac{1}{2} M_{12} + \frac{1}{(N-2)} \sum_{1 \leq i < j \leq N} D_{ij} \end{aligned} \quad (17)$$

S_{ij} can be transformed into the form of M_{ij} , thus the goal to minimize S_{ij} is equivalent to minimize M_{ij} .

Hence, the algorithm first computes $\{M_{ij}\}_{i \neq j}$ and chooses minimum M_{ij} to join nodes i and j . Then compute the new distance matrix from all other nodes to the new node joined. Iteratively join the nodes by pairs and update the new distance matrix with the new node until all nodes are joined together and the unrooted tree of minimum sum of branch length is formed.

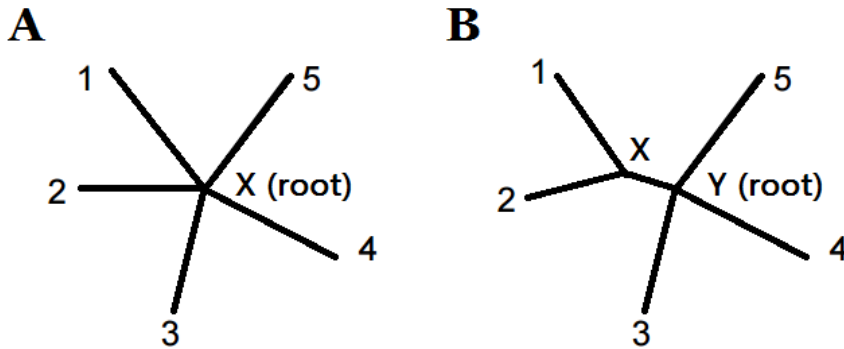


Figure 3. Neighbor-joining method

2.5 Allocate the Root of the Tree

The NJ method constructed an unrooted tree and we need to find the most likely root of the tree to infer the evolutionary order of the genes. The information is embedded in the relative size of MCPs we derived from the two-way Poisson mixture model.

The RankNet method integrates the pairwise information into a global ranking so that we can compare any subsets in a global point of view.

We first choose a point in the middle of one tree branch to be the root. Then, from the bottom of the tree, sequentially compare the rank of each leaf node. We push the higher ranked mutation upward to the closest common ancestor. For example, if mutation A in Figure 4 is ranked higher than mutation B, then we move mutation A upward to their most recently common ancestor node as panel 4A. In the next step, mutation A will be compared with mutation C. The place where a mutation stays at the end of these comparison steps is the inferred time of occurrence for this mutation. This position can only be one of its ancestral nodes in between the leaf it starts and the root. In the case where mutation B is ranked higher, B is moved upward to the higher level and then be compared with C (Figure 4B).

The arrangement in Figure 4A implies the MCPs of mutation A is usually larger than that of B across all the samples considered. Hence, it is very likely that mutation A occurs earlier than mutation B. According to the tree structure, mutation A and mutation B are closely related and it is very likely that mutation B coexists with mutation A in the same set of cells. That forms the subclone structures we would like to infer as demonstrated in Figure 4C. After the genes are allocated to places they appear, we used pairwise rank error rate to evaluate the tree. We pick the tree with minimum pairwise rank error rate to be our final evolution tree. The concept of

pairwise rank error rate is stated bellow.

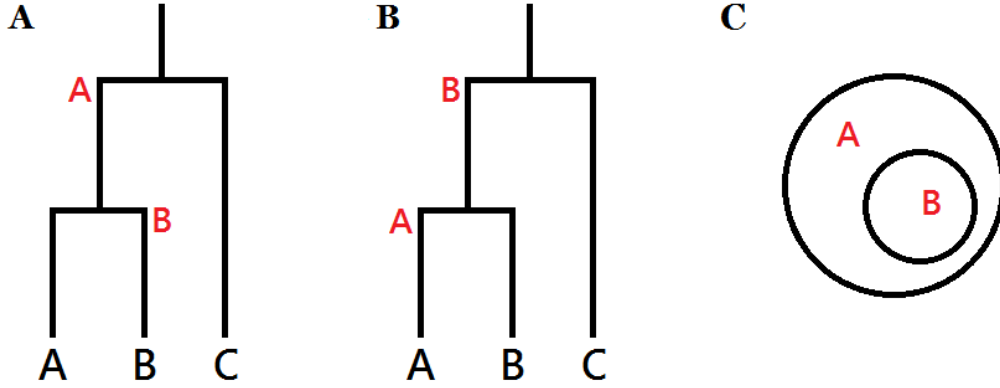


Figure 4. A toy example for how genes were pushed upwards. (A) A ranks higher than B. (B) B ranks higher than A. (C) Subclone structures of mutation A and B inferred in Figure 4A.

The criterion we use for choosing this optimal tree is minimum pairwise rank error. Since we know the length from the root to the interior node where the mutation of genes take place, all distance from root to mutation appearance time can be computed. The length of tree branches are proportional to the time mutations first appear. For a pair of mutation, say mutation A and B, if the distance of A to the root is longer than that of B to the root, but A has a higher ranking, we count this pair as an error. This happens when the two mutations are far apart on the tree that cannot be compared directly. We go through all the pairs of mutation, and get the pairwise rank error rate.

That is, if we have N mutations, $C_2^N = \frac{N(N-1)}{2}$ pairs of mutations would be compared to get the error rate.

For example (Figure 5), a neighbor-joining tree constructed from the MCP values across samples for mutations A, B, C, D is as Figure 5A. If the ranking order from RankNet is A, B, C, then D, and we picked the root in the middle of a particular

branch, and we pushed higher ranked mutations upward, the final evolutionary tree we constructed (Figure 5B) would have pairwise rank error rate equal to $\frac{2}{c_2^4} = 0.3333$, since (B, C) and (B, D) are of wrong order, and mutation B should be placed closer to the root.

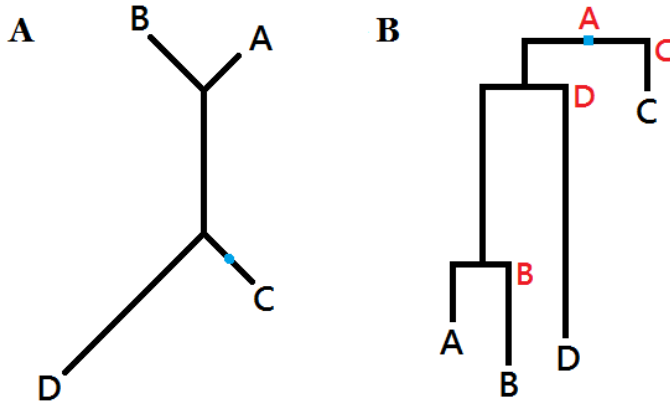


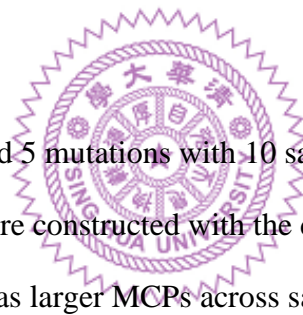
Figure 5. A toy example for calculating pairwise rank error rate. (A) A NJ tree derived from MCP values across samples. (B) Result of a particular root selected (the blue point). The pairwise rank error rate equals to 0.3333.

RankNet gets slightly different ranking orders due to the randomness of initial weights and the stochastic process of choosing a pair of instance in each iteration. To decide if one mutation is ranked higher than the other, we averaged the results of 100 replicated RankNet outputs to get a mean ranking order of the mutation of genes in a list. The rankings we use for comparison is the average rank for the mutation.

3. Results

To see how well the tree is constructed, we apply the reconstruction method to both simulated and real datasets. In the simulation, we construct trees of some known structures to see if the results can find both the correct rankings for the occurrence of gene mutations and the evolution relationships for those mutations. We also evaluate the reconstruction method by testing it on a head and neck cancer data of 32 samples with record of either invasion or metastasis from TCGA, to see if potential evolutionary orders or some functional groups and pathway of mutations can be identified.

3.1 Simulation



Our first toy example generated 5 mutations with 10 samples, and their MCPs are shown in Table 1. The MCPs are constructed with the designed clonal structure as in Figure 6. Mutation A always has larger MCPs across samples than mutation B. It implies that mutation A occurs earlier than mutation B and mutation B is included in a subset of cells with mutation A. We use NJ method to construct an unrooted tree according to their similarity of MCPs across samples (Figure 7A). From the average result of 100 RankNet trials, the ranking goes as follow: A, B, C, D, then E, and then we decide a root by minimizing pairwise ranking error after placing the higher ranked mutations upward (Figure 7B). In this example, the rank error rate is zero since no reversed rank pairs exist after the root is picked.

This result of the final tree is consistent to the ground truth relationships designed in Figure 6. Mutations B, C, D and E are all included in the subclone with the mutation

of A. That is, when mutation A take place, other mutations come afterwards.

Furthermore, E is included in B. This meets our design since for every sample having mutation B, mutation E is also included with a smaller size.

Table 1. A toy example

	sample 1	sample 2	sample 3	sample 4	sample 5	sample 6	sample 7	sample 8	sample 9	sample 10
A	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
B	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0	0	0
C	0.8	0.8	0.8	0.8	0	0	0	0.8	0.8	0.8
D	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
E	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0	0	0

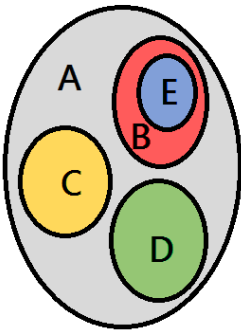


Figure 6. The clonal structure designed for the toy example.

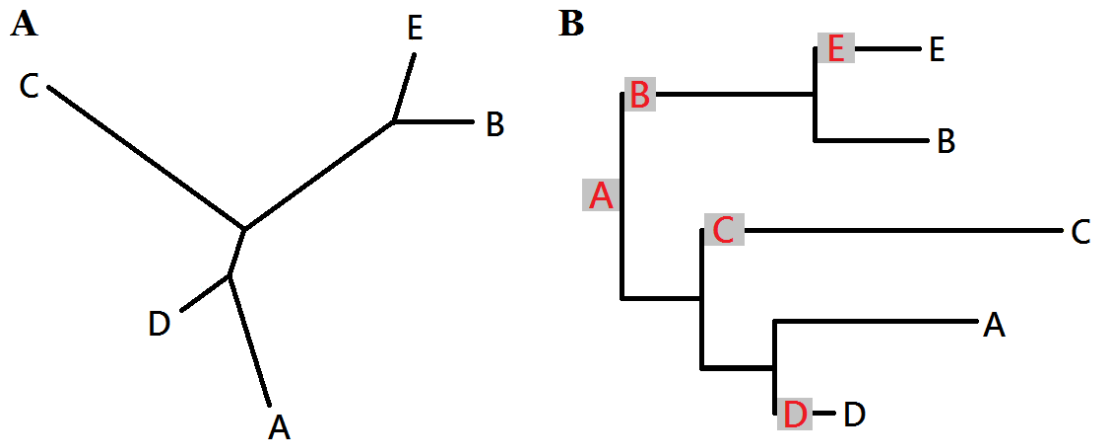


Figure 7. (A) The unrooted tree constructed by NJ method. (B) Result of the rooted tree with higher ranked mutations pushed upward to the place where that mutation appeared.

Our second simulation dataset generated 30 samples and 100 mutations from 10 mutation groups (Figure 8). It differs from the first toy example in considering the estimation errors caused by the two-way Poisson mixture model. The 100 variants belong to 10 groups of mutation patterns. For example, mutations in group A are likely to be followed by mutations in group B or, in other words, to cause the mutations in group B to happen. Similarly, mutations in group C, D and E take place in part of the samples with mutations of group B. Then mutations in group E is likely to be followed by group F mutations. Figure 9 shows the designed clonal structure of the ten groups of mutations.

Some of the MCP values estimated from the two-way Poisson mixture model are zero, which represents samples without the mutations. Hence, we assigned certain proportion of the samples to get zero MCPs. For those non-zero MCPs, we generated the values with a normal distribution for each gene group. MCP values generated in the simulation are plotted in Figure 8 as a heat map with higher values in red and lower values in blue, and the distributions are listed in Table 2. For example, type A mutation has MCP values generated from the normal distribution with mean 0.8 and a standard deviation of 0.2. If the value generated is above 1 or below 0, we truncate those values by 0.999999 and 0 respectively.

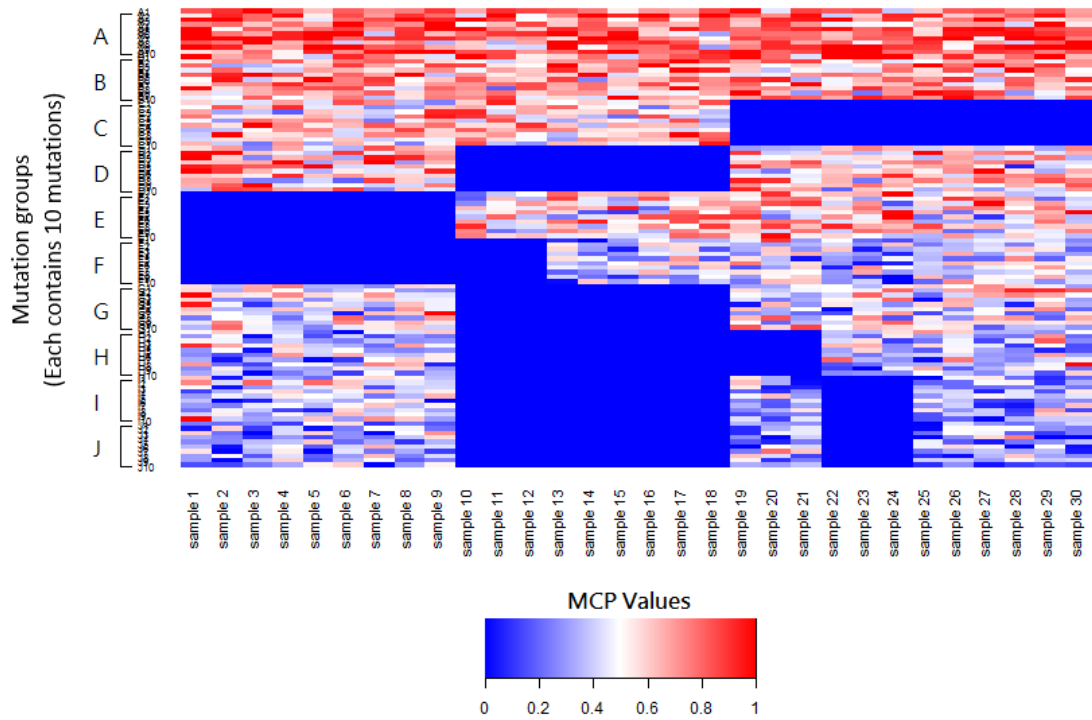


Figure 8. Heatmap of the simulated data of 100 mutations across 30 samples with MCP values closer to 1 colored in red and values closer to 0 colored in blue.

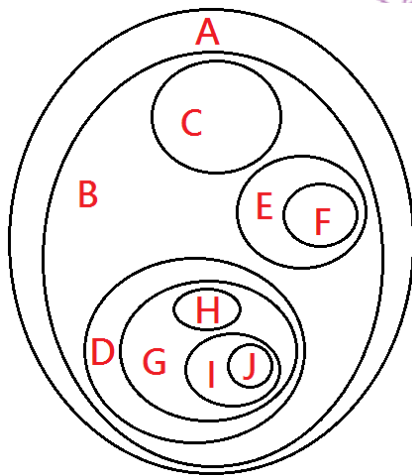
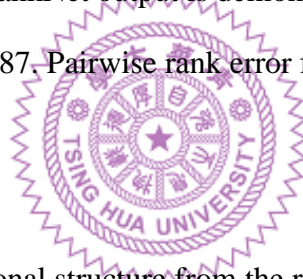


Figure 9. Illustration of the designed clonal structure of relationships for the simulation data.

Table 2. Distribution of MCP values for each mutation group

Mutation group	Distribution
A	$Normal(\mu = 0.8, \sigma = 0.2)$
B	$Normal(\mu = 0.7, \sigma = 0.2)$
C	$Normal(\mu = 0.6, \sigma = 0.2)$
D	$Normal(\mu = 0.6, \sigma = 0.2)$
E	$Normal(\mu = 0.6, \sigma = 0.2)$
F	$Normal(\mu = 0.4, \sigma = 0.2)$
G	$Normal(\mu = 0.5, \sigma = 0.2)$
H	$Normal(\mu = 0.4, \sigma = 0.2)$
I	$Normal(\mu = 0.4, \sigma = 0.2)$
J	$Normal(\mu = 0.3, \sigma = 0.2)$

The unrooted tree derived from the Neighbor-joining method is shown in Figure 10A. Mutations of the same group are mostly clustered together. Groups with similar MCP levels are also closely clustered such as group A and group B. The rooted tree constructed according to the RankNet output is demonstrated in Figure 10B and the pairwise rank error rate is 0.3987. Pairwise rank error for each branch is shown in Figure A1 in the appendix.



To better understand the subclonal structure from the rooted tree in Figure 10B, we first selected the mutations with the largest rank within each group on the rooted tree and then used their phylogenetic relation to reconstruct the subclonal structure in Figure 11. We can compare this to the structure designed in Figure 9. All of the differences are towards the bottom of the rooted tree. For example, group E mutations are designed to contain group F mutations, but mutation E8 contains all other mutations except for A10 and B1. Type G mutations are supposed to trigger mutation groups H, I and J, but mutation G10 did not contain mutations H4, I2 and J8 after it appeared on the rooted tree. All the other relationships are quite consistent to the desired structure. This is a simple consequence of insufficient information since those mutations have small cellular occupancy according to our generation procedure in

Table 2. This also leads to a less distinguishable cluster mixed with group D and G as well the cluster mixed with I and J in Figure 10B.

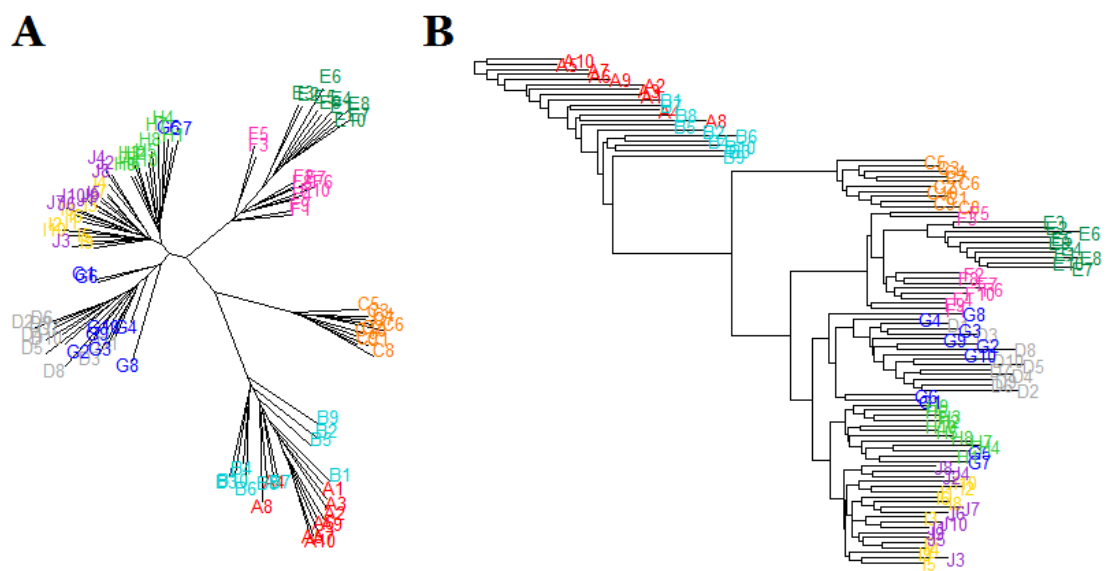


Figure 10. (A) The unrooted tree constructed by NJ method with mutations in the same group labeled by the same color for simulation data. (B) The rooted tree constructed by choosing a root with minimum pairwise rank error rate. Mutations of the same group have the same color labels.

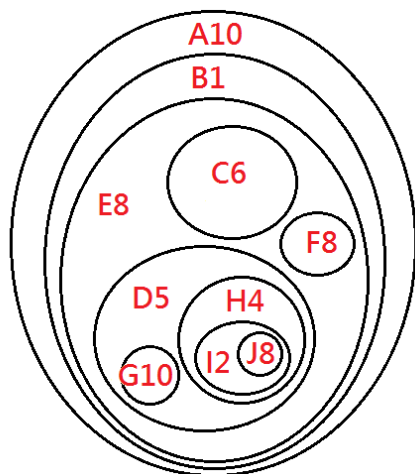


Figure 11. A subgraph of clonal structure of the rooted tree in Figure 9B. Only the mutation ranked the highest in each group is shown.

For the top part of the tree, their evolutionary orders are perfectly reconstructed.

Figure 12 shows the top 15 mutations that are close to the root of the tree in Figure 10B. They are mostly mutations from group A and B. Their average ranks from the 100 RankNet outputs are listed in Table 3 and the rank is highly correlated to the distance from the root. There are some inversed ranking since mutations not pushed up to the node of their common ancestor will not be compared. The results showed that our tree construction strategy can almost restore the evolutionary order of the early mutations and it is important for identifying the disease causing genes that occurred at the very early stage of the tumor development.

Table 3. Average rank for the top 15 genes on the tree from simulation data.

order of the distance from the root	gene	rank	order of the distance from the root	gene	rank
1	A10	1.00	9	A1	8.85
2	A7	2.22	10	A8	9.98
3	A5	2.90	11	B7	11.79
4	A3	4.33	12	A4	15.58
5	A6	6.04	13	B10	11.09
6	A9	4.57	14	B8	13.48
7	A2	7.65	15	B5	13.55
8	B1	7.46			

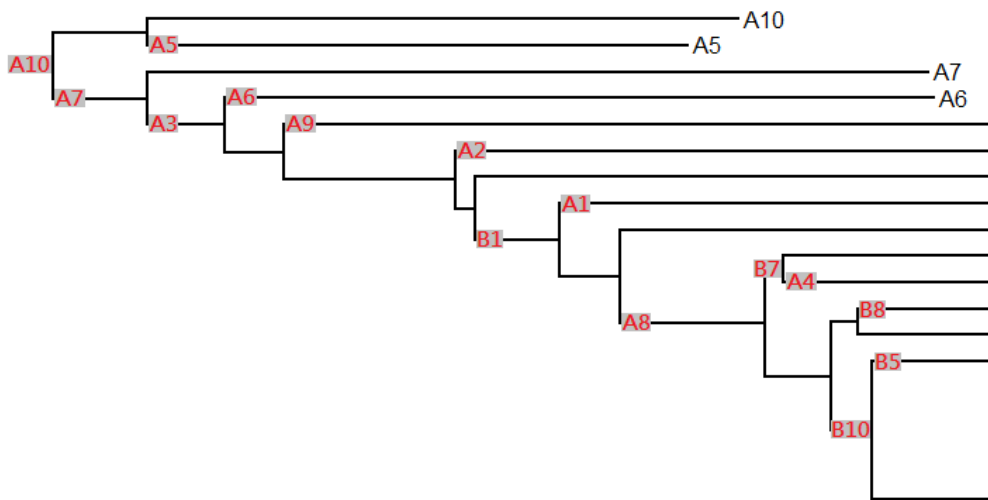


Figure 12. Top part of the tree for simulation data (15 genes)

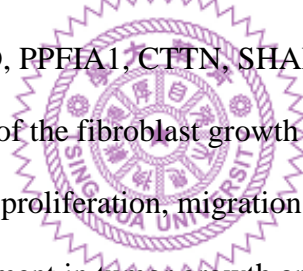
3.2 On real dataset

As mentioned in section 2.1, the dataset contains 32 samples and 3,244 mutation of genes. After we picked a root that minimize rank error rate, the results are shown below. Pairwise rank error rate for each branch is shown in Figure A2 in the appendix with minimum error rate of 0.3901. Table 4 is the top 15 genes that appear on the tree, and their average rank is listed on the right. We can see that only TFG has rank higher than 1000, the other genes have ranks lower than 600. TFG comes 4-th place in the order of appearance since the leave it started is pulled very near the root, when minimum pairwise rank error rate root was picked. Thus, no matter how large the rank is, it only has places near the root to pick for the time it appear.

Figure 13 shows the top 15 mutations that are close to the root of the final tree we constructed. SCR1B can be interpreted as the mutation from the founder cell. The mutation of gene PCCB is likely to follow the mutation in SCR1B, and the mutation of TFG comes afterwards. Also on the other side branch, gene SLC9A9 is likely to mutate after SCR1B, and then ESYT3, PLOD2 and STAG1 sequentially follows.

Table 4. Average rank for the top 15 genes on the tree from real data.

order of the distance from the root	gene	rank	order of the distance from the root	gene	rank
1	SCR1B	3.50	9	PLOD2	400.50
2	PCCB	446.91	10	CCDC37	300.08
3	SLC9A9	47.00	11	ISY1	530.66
4	TFG	1282.83	12	STAG1	55.16
5	ESYT3	597.28	13	ASL	360.58
6	ADAC9	128.58	14	DZIP1L	583.00
7	PIK3CB	80.25	15	KTELC1	573.16
8	C3orf17	541.50			



Previous studies have found that chromosome 11q13 is amplified frequently in head and neck squamous cell carcinoma and it includes FGF4, TPCN2, MYEOV, CCND1, ORAOV1, TMEM16A, FADD, PPFIA1, CTTN, SHANK2 and DHCR7 (Sugahara K. et al.)[12]. FGF4 is a member of the fibroblast growth factor (FGF) family and it has diverse roles in regulating cell proliferation, migration and differentiation activities. It has been proposed for involvement in tumor growth and lymph node metastasis [13]. Since cell migration is a crucial step during metastatic cascade, we evaluate the mutation of genes in this region.

28

FGF4, SAPS3, SLC29A2, SHANK2, TPCN2, FGF19, CTTN, MYEOV, FGF3, ORAOV1, PPFIA1, and FADD. We can see that the downstream mutations of CCND1 are mostly mutations on region 11q13.

Furthermore, we use Gene Set Enrichment Analysis (GSEA) to derive the significantly enriched Gene ontology terms for those common ancestors (Table 5) as well as the genes on chromosome 11q13. GO term analysis result is shown in Table 6. There are eight significant genesets with FDR q-values less than 0.05. The most significant GO term is related to cell junctions, which is important in enabling communication between neighboring cells. Adhesion, synapse, anchoring junction and negative regulation of cell death are also biological functions related to metastasis of tumors. From the evolution tree constructed, we may be able to associate mutations to some clinical symptoms of diseases.

The genes with each of the eight GO terms are listed in the Appendix Figure A3. Only ORAOV1 and MYEOV in 11q13 region are not attached with the significant GO terms. Some other genes in the ancestor list such as SCRIB are related to seven GO terms; SDK1 is related to three GO terms and probably harvest important mutation related to head and neck cancer.

Table 5. Common ancestors for all the genes in 11q13 region except for DHCR7. Genes marked in grey will not be included if DHCR7 is considered.


Common Ancestors For All Genes	SCRIB	ESYT3	SLC9A9	PCCB
	ACAD9	PIK3CB	ISY1	DZIP3
	SYCP2	NUP155	KIAA1797	XPO4
	KIAA0020	CHL1	SAMHD1	ANO1
	COL9A3	CTSW	LMBRD2	SDK1
	ARHGAP26	PLEKHG4B	CCND1	SART1
	ADRBK1	NADSYN1	CBX4	DLGAP4

Table 6. GO term analysis result for the upstream mutations of genes in 11q13 region.

GO terms [# Genes (K)]	Number of Genes (k)	p-value	FDR q-value
CELL JUNCTION [1151]	8	1.95 e ⁻⁶	1.15 e ⁻²
ADHESION [1032]	7	1.1 e ⁻⁵	3.24 e ⁻²
SYNAPSE [754]	6	2.07 e ⁻⁵	4.08 e ⁻²
ANCHORING JUNCTION [489]	5	3.33 e ⁻⁵	4.57 e ⁻²
NEURON PART [1265]	7	4.03 e ⁻⁵	4.57 e ⁻²
NEGATIVE REGULATION OF CELL DEATH [872]	6	4.65 e ⁻⁵	4.57 e ⁻²
RESPONSE TO EXTERNAL STIMULUS [1821]	8	5.41 e ⁻⁵	4.57 e ⁻²
APOPTOTIC PROCESS INVOLVED IN MORPH RPHOGENESIS [16]	2	6.72 e ⁻⁵	4.97 e ⁻²

4. Conclusion and Discussion

We proposed a procedure to reconstruct the evolutionary tree of tumor mutations regarding to the copy number aberrations. It is a procedure combined with the two-way Poisson mixture model, the Neighbor-joining tree and the RankNet algorithm. In this thesis, what we have done is, from the MCPs of multiple samples, to reconstruct a tree that contains both the information of mutation occurrence time, and to infer the relationships for those mutations. That is, we can see from the tree that for mutations closer to the root, it occurred earlier in time, and we can also tell the probable mutation that triggers a certain mutation and also the possible offspring that comes after the mutation. Moreover, the structure of cellular evolution can be seen.



Our tree shows how the mutations evolve on average across multiple samples. That is, probably most of the time one mutation follows by its progeny mutation gene on the tree, but it does not guarantee that the mutation orders will be followed for every single sample. Although we do not provide the clonal structure of every single tumor, it is still a good reference for the discovery of founder gene and how these gene mutations interact.

In the use of NJ method, Euclidean distance is for the convenience of making equal weight to all samples. If we have the information of which sample is more representative for a particular type of head and neck cancer, we can improve the tree construction by considering other weighted distance between genes. Furthermore, the distance contains both information of the time and relationships (clonal structural

difference of genes across samples) of mutations. That is, the branch length of our Neighbor-joining tree contains two types of information, and we cannot separate them.

RankNet identifies the copy number mutations that occur early in the tumor stage. It sorts out all 3,244 mutations occurrence time by only input pairwise comparison of those genes. Genes on the top of the list having high MCP values among all samples would be good candidates for disease driver genes.

Since RankNet is based on the neural network model, there are a lot of parameters that we can regulate and control. The number of hidden layers we used is one and the number of nodes in the hidden layer is ten in our study. Due to the time limit, we are not able to carry out the comparison across different conditions. RankNet might be more robust with different settings. By adding more hidden layers or adding nodes in the hidden layers might solve the consistency problem of RankNet among repeated trials. In addition, choosing different learning rate and max number of iterations may improve the computational efficiency of the algorithm. It will require a more comprehensive simulation to assess those proposals.

In determining where the root of the tree should be, the criterion we adopt is to minimize the pairwise rank error rate. For each pair of mutations on the tree, we check if their relative size of distance from the root is consistent to their ranking under the RankNet output and then we calculate the proportion of pairs that resulted in reverse order as the error rate. A different approach is to check if most of the samples have MCPs following the same order of the distance to the root. If less than half of the samples follow the order, we count it as an error pair.

Since the phylogeny tree we construct is a binary tree, it implies that each clone can only have at most two subclones. If two or more mutation (sets) occur in parallel are around the same time, it will not be correctly interpreted in our procedure. This is actually a common issue to the phylogeny tree approach. At the step of finding the root, we push up one of the two genes with the common ancestor. If we have a fair criterion to retain both at the same place, we can have an ancestor followed with more than two descendants. However, the interpretation of an empty inner node would be another problem.

Our method for the construction of the evolutionary tree is a two-step method. We first find an unrooted NJ tree and then we put the information of occurrence order from RankNet in to the tree to find a root as well as to infer the place where each mutation takes place on the tree. If there is a way that we can combine both the structural information of the tree together with the order of occurrence time of all mutations, and construct the evolutionary tree with these two types of information at the same time, it would be a better conclusion for phylogenetic tree reconstruction for cancer disease.

Finally, the most important issue to be addressed is to understand the relationship between tumor heterogeneity and the disease progression. Our method constructed an evolutionary tree inferring the relationship of gene mutations. It can be used to see where current mutation in the evolutionary hierarchy is so that we can find out probable genes that trigger the mutation. They can be good target for disease treatments.

Reference

1. Nowell, P.C., *The clonal evolution of tumor cell populations*. Science, 1976. **194**(4260): p. 23-28.
2. Tai, A.-S., et al. (2017), *A two-way mixture model towards the decomposition of tumor heterogeneity*. Unpublished doctoral dissertation, National Tsing Hua University, Hsinchu, Taiwan.
3. Hanahan, D. and R.A. Weinberg, *The hallmarks of cancer*. cell, 2000. **100**(1): p. 57-70.
4. Liu, T.-Y., *Learning to rank for information retrieval*. Foundations and Trends® in Information Retrieval, 2009. **3**(3): p. 225-331.
5. Li, H., *A short introduction to learning to rank*. IEICE TRANSACTIONS on Information and Systems, 2011. **94**(10): p. 1854-1862.
6. Burges, C., et al. *Learning to rank using gradient descent*. in *Proceedings of the 22nd international conference on Machine learning*. 2005. ACM.
7. Burges, C.J., *From ranknet to lambdarank to lambdamart: An overview*. Learning, 2010. **11**(23-581): p. 81.
8. Sneath, P.H. and R.R. Sokal, *Numerical taxonomy. The principles and practice of numerical classification*. 1973.
9. Saitou, N. and M. Nei, *The neighbor-joining method: a new method for reconstructing phylogenetic trees*. Molecular biology and evolution, 1987. **4**(4): p. 406-425.
10. Peng, C., *Distance based methods in phylogenetic tree construction*. NEURAL PARALLEL AND SCIENTIFIC COMPUTATIONS, 2007. **15**(4): p. 547.

11. Studier, J.A. and K.J. Keppler, *A note on the neighbor-joining algorithm of Saitou and Nei*. Molecular biology and evolution, 1988. **5**(6): p. 729-731.
12. Sugahara, K., et al., *Combination effects of distinct cores in 11q13 amplification region on cervical lymph node metastasis of oral squamous cell carcinoma*. International journal of oncology, 2011. **39**(4): p. 761-769.
13. Muller, D., et al., *Frequent amplification of 11q13 DNA markers is associated with lymph node involvement in human head and neck squamous cell carcinomas*. European Journal of Cancer Part B: Oral Oncology, 1994. **30**(2): p. 113-120.



Appendix

RankNet Algorithm

Set number of hidden node L , initial weights w , offsets b and learning rate η

for $t = 0$ to T **do**

1. stochastic: randomly pick a QDP pair with feature vector, say $\{x_i, x_j\}$
2. forward: compute $y_i = f(x_i)$ and $y_j = f(x_j)$ with current model parameters $w^{(t)}$ and $b^{(t)}$, then compute the loss $C_{ij}^{(t)}$
3. backward: compute $\frac{\partial C_{ij}^{(t)}}{\partial w^{(t)}}$ and $\frac{\partial C_{ij}^{(t)}}{\partial b^{(t)}}$
4. gradient descent: update the parameters $w \leftarrow w - \eta \cdot \frac{\partial C}{\partial w}$ and

$$b \leftarrow b - \eta \cdot \frac{\partial C}{\partial b}$$

end for

return $f(\cdot)$



Initial values and parameters used for RankNet

hidden nodes	$L = 10$
initial weights	$w^{(0)} \sim Unif(-0.1, 0.1)$
initial offsets	$b^{(0)} \sim Unif(-0.1, 0.1)$
learning rate	$\eta = 1 \times 10^{-2}$
max iteration number	$T = 8000$

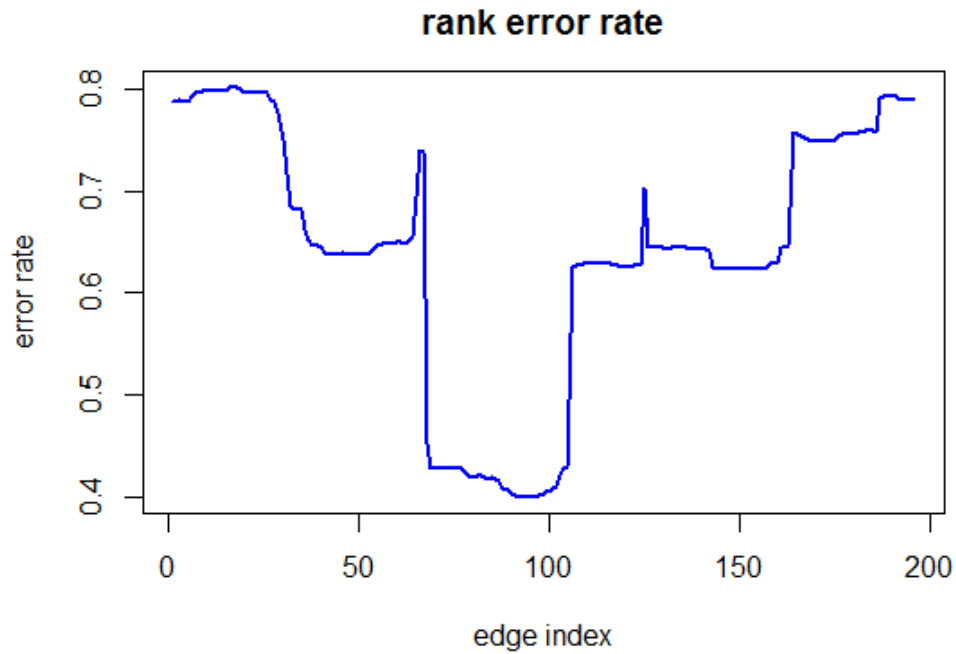


Figure A1. Pairwise rank error rate for simulated data set. The edge index is the label of branches generated from NJ function in the R package ‘ape’. Minimum error rate = 0.3987

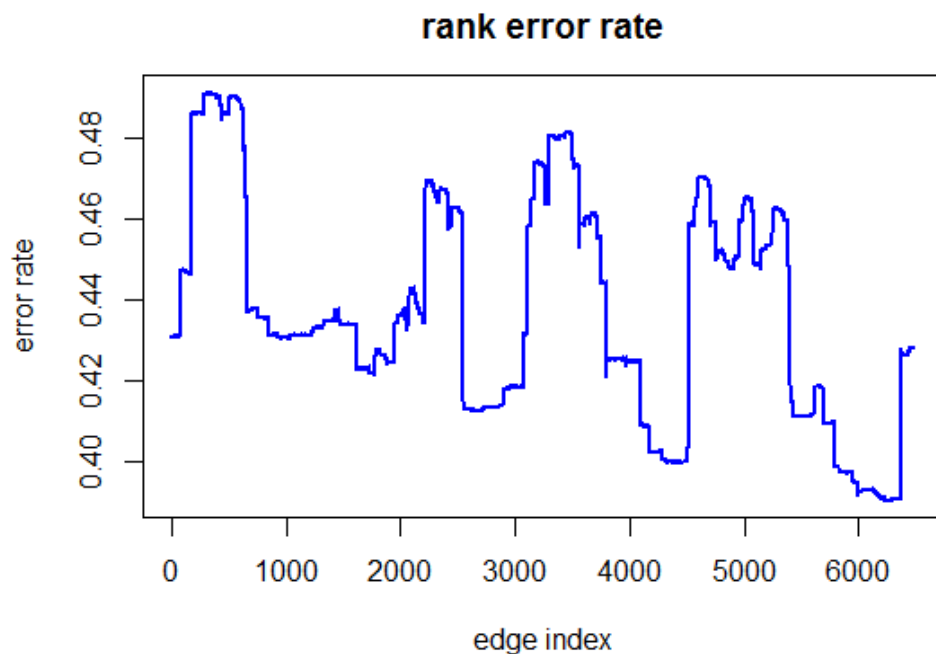


Figure A2. Pairwise rank error rate for the real head and neck cancer data set. The edge index is the label of branches generated from NJ function in the R package ‘ape’. Minimum error rate = 0.3901

Gene	CELL JUNCTION	ADHESION	SYNAPSE	ANCHORING JUNCTION	NEURON PART	NEGATIVE REGULATION OF CELL DEATH	RESPONSE TO EXTERNAL STIMULUS	APOPTOTIC PROCESS INVOLVED IN MORPH	Gene Description
CTTN									cortactin
SCRIB									scribbled homolog (Drosophila)
PPFIA1									protein tyrosine phosphatase, receptor type, f polypeptide (PTPRF), interacting protein (liprin), alpha 1
SDK1									sidekick homolog 1, cell adhesion molecule (chicken)
SHANK2									SH3 and multiple ankyrin repeat domains 2
ARHGAP26									Rho GTPase activating protein 26
KIAA1797									KIAA1797
CCND1									cyclin D1
CHL1									cell adhesion molecule with homology to L1CAM (close homolog of L1)
FADD									Fas (TNFRSF6)-associated via death domain
PIK3CB									phosphoinositide-3-kinase, catalytic, beta polypeptide
DLGAP4									discs, large (Drosophila) homolog-associated protein 4
ACAD9									acyl-CoA dehydrogenase family, member 9
FGF4									fibroblast growth factor 4
SYCP2									synaptonemal complex protein 2
CBX4									chromobox homolog 4
TPCN2									two pore segment channel 2
SAMHD1									SAM domain and HD domain 1
ANO1									anoctamin 1, calcium activated chloride channel
COL9A3									collagen, type IX, alpha 3
DZIP3									DAZ interacting protein 3, zinc finger
PCCB									propionyl CoA carboxylase, beta polypeptide
NADSYN1									NAD synthetase 1
ADRBK1									adrenergic, beta, receptor kinase 1
NUP155									nucleoporin 155kDa
XPO4									exportin 4
SART1									squamous cell carcinoma antigen recognized by T cells
ISY1									ISY1 splicing factor homolog (S. cerevisiae)
CTSW									cathepsin W
KIAA0020									KIAA0020
SLC9A9									solute carrier family 9 (sodium/hydrogen exchanger), member 9
ORAOV1									oral cancer overexpressed 1
ESYT3									extended synaptotagmin-like protein 3
MYEOV									myeloma overexpressed (in a subset of t(11;14) positive multiple myelomas)
LMBRD2									LMBR1 domain containing 2

Figure A3. Genes enriched in each of the eight significant GO terms. Genes with yellow background are genes in 11q13 region.