

# Detecting the correlated mutations based on selection pressure with CorMut

Zhenpeng Li

October 29, 2019

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Methods</b>	<b>2</b>
<b>3</b>	<b>Implementation</b>	<b>3</b>

## 1 Introduction

In genetics, the Ka/Ks ratio is the ratio of the number of non-synonymous substitutions per non-synonymous site (Ka) to the number of synonymous substitutions per synonymous site (Ks)[1]. Ka/Ks ratio was used as an indicator of selective pressure acting on a protein-coding gene. A Ka/Ks ratio of 1 indicates neutral selection, i.e., the observed ratio of non-synonymous mutations versus synonymous mutations exactly matches the ratio expected under a random mutation model. Thus, amino acid changes are neither being selected for nor against. A Ka/Ks value of  $<1$  indicates negative selection pressure. That is to say most amino acid changes are deleterious and are selected against, producing an imbalance in the observed mutations that favors synonymous mutations. In the condition of  $Ka/Ks > 1$ , it indicates that amino acid changes are favored, i.e., they increase the organism's fitness. This unusual condition may reflect a change in the function of a gene or a change in environmental conditions that forces the organism to adapt. For example, highly variable viruses mutations which confer resistance to new antiviral drugs might be expected to undergo positive selection in a patient population treated with these drugs, such as HIV and HCV.

The concept of correlated mutations is one of the basic ideas in evolutionary biology. The amino acid substitution rates are expected to be limited by functional constraints. Given the functional constraints operating on gene, a mutation in one position can be compensated by an additional mutation. Then mutation patterns can be formed by correlated mutations responsible for specific conditions.

Here, we developed an R/Bioconductor package to detect the correlated mutations among positive selection sites by combining ka/ks ratio and correlated mutations analysis. The definition of Ka/Ks is based on Chen et al[2]. CorMut provides functions for computing kaks for individual site or specific amino acids and detecting correlated mutations among them. Two methods are provided for detecting correlated mutations,

including conditional selection pressure and mutual information, the computation consists of two steps: First, the positive selection sites are detected; Second, the mutation correlations are computed among the positive selection sites. Note that the first step is optional. Meanwhile, CorMut facilitates the comparison of the correlated mutations between two conditions by the means of correlated mutation network.

## 2 Methods

### 2.1 Ka/Ks calculation for individual codon position and specific amino acid substitutions

The Ka/Ks values for individual codon (specific amino acid substitutions) were determined as described by Chen et al. The Ka/Ks of individual codon (specific amino acid substitution (X2Y) for a codon) was computed as follows:

$$\frac{K_a}{K_s} = \frac{\frac{N_Y}{N_S}}{\frac{n_{Y,t}f_t + n_{Y,v}f_v}{n_{S,t}f_t + n_{S,v}f_v}}$$

where NY and NS are the count of samples with nonsynonymous mutations(X2Y mutation) at that codon and the count samples with of synonymous mutations observed at that codon respectively. Then, NY/NS is normalized by the ratio expected under a random mutation model (i.e., in the absence of any selection pressure), which was represented as the denominator of the formula. In the random mutation model, ft and fv indicate the transition and transversion frequencies respectively, and they were measured from the entire data set according to the following formulas: ft = Nt/ntS and fv = Nv/nvS, where S is the total number of samples; Nt and Nv are the numbers of observed transition and transversion mutations, respectively; nt and nv are the number of possible transitions and transversions in the focused region (simply equal to its length L and 2L respectively).LOD confidence score for a codon or mutation X2Y to be under positive selection pressure was calculated by the following formula:

$$LOD = -\log_{10}p(i \geq N_{YaXa}|N, q, \left(\frac{K_a}{K_s}\right)_{Y|Xa} = 1) = -\log_{10} \sum_{i=N_{YaXa}}^N \binom{N}{i} q^i (1-q)^{N-i}$$

Where N = NYaXa + NYsXa and q as defined above. If LOD > 2, the positive selection is significant.

### 2.2 Mutual information

MI content was adopted as a measure of the correlation between residue substitutions[3]. Accordingly, each of the N columns in the multiple sequence alignment generated for a protein of N codons is considered as a discrete random variable  $X_i(1 \leq i \leq N)$ . When compute Mutual information between codons,  $X_i(1 \leq i \leq N)$  takes on one of the 20 amino acid types with some probability, then compute the Mutual information between two sites for specific amino acid substitutions,  $X_i(1 \leq i \leq N)$  were only divided two parts (the specific amino acid mutation and the other) with some probability. Mutual information describes the mutual dependence of the two random variables  $X_i, X_j$ . The MI associated the random variables  $X_i$  and  $X_j$  corresponding to the ith and jth columns is defined as

$$I(X_i, X_j) = S(X_i) + S(X_j) - S(X_i, X_j) = S(X_i) - S(X_i|X_j)$$

Where

$$S(X_i) = - \sum_{all x_i} P(X_i = x_i) \log P(X_i = x_i)$$

is the entropy of  $X_i$ ,

$$S(X_i|X_j) = - \sum_{all x_i} \sum_{all x_j} P(X_i = x_i, X_j = x_j) \log P(X_i = x_i|X_j = x_j)$$

is the conditional entropy of  $X_i$  given  $X_j$ ,  $S(X_i, X_j)$  is the joint entropy of  $X_i$  and  $X_j$ . Here  $P(X_i = x_i, X_j = x_j)$  is the joint probability of occurrence of amino acid types  $x_i$  and  $x_j$  at the  $i$ th and  $j$ th positions, respectively,  $P(X_i = x_i)$  and  $P(X_j = x_j)$  are the corresponding singlet probabilities.

$I(X_i, X_j)$  is the  $ij$ th element of the  $N \times N$  MI matrix  $I$  corresponding to the examined multiple sequence alignment.

### 2.3 Jaccard index

Jaccard index measures similarity between two variables, and it has been widely used to measure the correlated mutations. For a pair of mutations  $X$  and  $Y$ , the Jaccard index is calculated as  $N_{xy}/(N_{xy}+N_{x0}+N_{y0})$ , where  $N_{xy}$  represents the number of sequences where  $X$  and  $Y$  are jointly mutated,  $N_{x0}$  represents the number of sequences with only  $X$  mutation, but not  $Y$ , and  $N_{y0}$  represents the sequences with only  $Y$  mutation, but not  $X$ . The computation was deemed to effectively avoid the exaggeration of rare mutation pairs where the number of sequences without either  $X$  or  $Y$  is very large.

## 3 Implementation

3.1 Process the multiple sequence alignment files *seqFormat* replace the raw bases with common bases and delete the gaps according to the reference sequence. As one raw base has several common bases, then a base causing amino acid mutation will be randomly chosen. Here, HIV protease sequences of treatment-naive and treatment will be used as examples.

```
> library(CorMut)
> examplefile=system.file("extdata", "PI_treatment.aln", package="CorMut")
> examplefile02=system.file("extdata", "PI_treatment_naive.aln", package="CorMut")
> example=seqFormat(examplefile)
> example02= seqFormat(examplefile02)
```

### 3.2 Compute kaks for individual codon

```
> result=kaksCodon(example)
> fresult=filterSites(result)
> head(fresult)
```

	mutation	kaks	lod	freq
1	3	150.500000	52.827378	1.0000
2	10	8.500000	24.207139	0.7367
3	12	15.000000	5.106647	0.0967
4	19	2.538855	3.208047	0.1167
5	24	3.233179	3.131694	0.1000
6	35	4.300086	4.600433	0.2633

### 3.3 Compute kaks for individual amino acid mutation

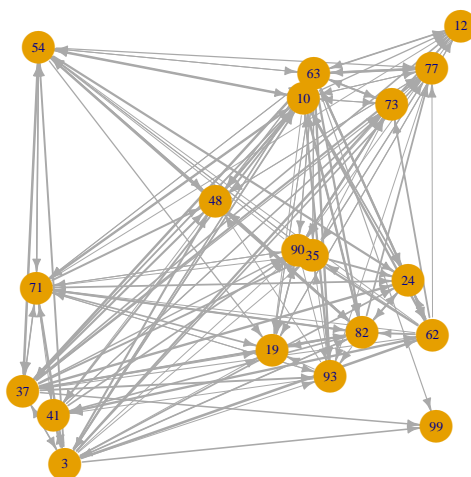
```
> result=kaksAA(example)
> fresult=filterSites(result)
> head(fresult)
```

	position	mutation	kaks	lod	freq
1	3	V3I	404.841453111799	179.656919924817	0.9967

2	10	L10I	106.510984639863	194.782739610876	0.6033
3	12	T12N	6.45113051800455	Inf	0.01
4	12	T12P	61.1996817820207	4.49416430538097	0.0267
5	12	T12E	15.0526378753439	Inf	0.0233
6	12	T12S	38.2498011137629	2.04605159081416	0.0167

### 3.4 Compute the conditional kaks(conditional selection pressure) among codons

```
> result=ckaksCodon(example)
> plot(result)
```



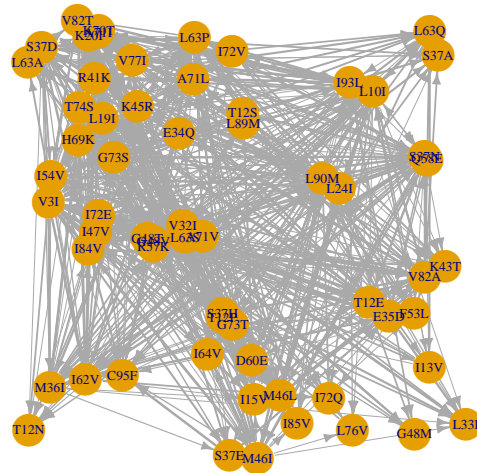
```
> fresult=filterSites(result)
> head(fresult)
```

	position1	position2	ckaks	lod
1	3	10	17	24.2071393392468
2	3	12	30	5.10664651261475
3	3	19	1.84210526315789	3.20804706405523
4	3	24	3.33333333333333	3.13169395564291
5	3	35	13.1666666666667	4.6004332373075
6	3	37	142	31.4550748700319

### 3.5 Compute the conditional kaks(conditional selection pressure) among amino acid mutations

```
> result=ckaksAA(example)
```

```
> plot(result)
```



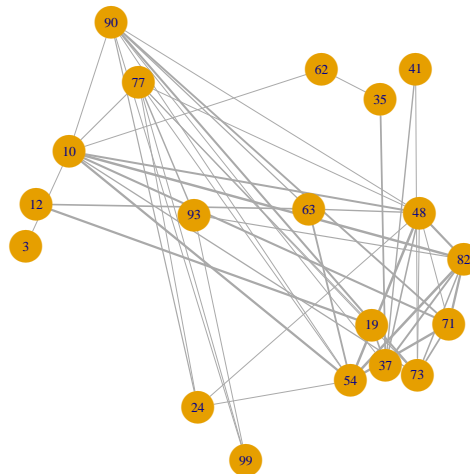
```
> fresult=filterSites(result)
> head(fresult)
```

	position1	position2	ckaks	lod
1	V3I	L10I	13.92	226.78
2	V3I	T12N	4	Inf
3	V3I	T12P	9	10.89
4	V3I	T12E	8	Inf
5	V3I	T12S	6	6.8
6	V3I	I13V	33	18.68

### 3.6 Compute the mutual information among codons

An instance with two matrixes will be returned, they are mutation information matrix and p matrix, The  $ij$ th element of the  $N \times N$  MI matrix indicates the mutation information or p value between position  $i$  and position  $j$ .

```
> result=miCodon(example)
> plot(result)
```



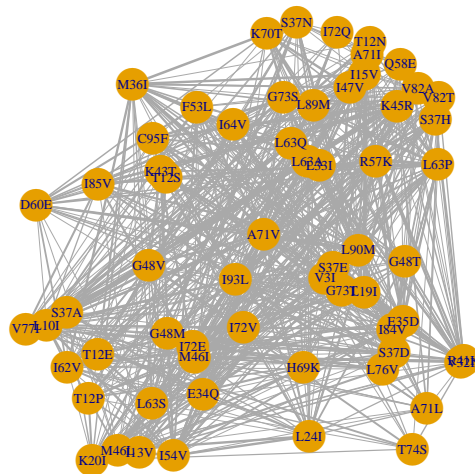
```
> fresult=filterSites(result)
> head(fresult)
```

	position1	position2	mi	p.value
1	10	3	0.0223403798434836	0.012948071418836
2	19	12	0.128809592522988	0.00371553353757904
3	37	19	0.0747160966532552	0.0192278860569715
4	37	35	0.075806272221832	0.00371553353757904
5	41	37	0.06512044936218	0.00371553353757904
6	48	10	0.0897750853050767	0.00371553353757904

### 3.7 Compute the mutual information among individual amino acid mutations

An instance with two matrixes will be returned, they are mutation information matrix and p matrix, The  $ij$ th element of the  $N \times N$  MI matrix indicates the mutation information or p value between two amino acid mutations.

```
> result=miAA(example)
> plot(result)
```



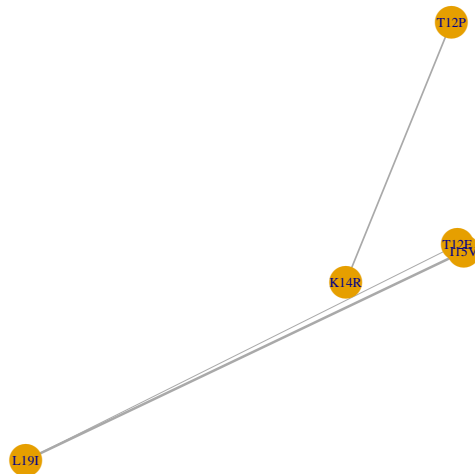
```
> fresult=filterSites(result)
> head(fresult)
```

	position1	position2		mi	p.value
1	L10I	V3I	0.00309067986573841	7.90929048032942e-40	
2	T12N	L10I	8.64708687707827e-05	0.040207170030247	
3	T12E	L10I	0.000640519978094889	1.63583633828262e-09	
4	I13V	V3I	0.000389134320368023	3.1062662483976e-05	
5	I13V	T12S	0.00478659729991238	6.31583852904427e-63	
6	I15V	V3I	0.00051661973913858	7.31371985853487e-07	

### 3.8 Compute the Jaccard index among individual amino acid mutations

An instance with two matrixes will be returned, they are Jaccard index matrix and p matrix, The  $ij$ th element of the  $N \times N$  MI matrix indicates the Jaccard index or p value between two amino acid mutations.

```
> result=jiAA(example)
> plot(result)
```



```
> fresult=filterSites(result)
> head(fresult)
```

	position1	position2	JI	p.value
1	T12P	K14R	0.192307692307692	0.0203598174209213
2	L19I	I15V	0.228070175438596	0.00322947408828938
3	T12E	I15V	0.162790697674419	0.000394448679750347
4	T12E	L19I	0.172413793103448	0.0203598174209213

### 3.9 Comparison of the correlated mutations between two conditions by the means of correlated mutation network

*biCompare* compare the correlated mutation relationship between two conditions, such as HIV treatment-naive and treatment. The result of *biCompare* can be visualized by *plot* method. Only positive selection codons or amino acid mutations were displayed on the plot, blue nodes indicate the distinct positive codons or amino acid mutations of the first condition, that is to say these nodes will be non-positive selection in second condition. While red nodes indicate the distinct positive codons or amino acid mutations appeared in the second condition. *plot* also has an option for displaying the unchanged positive selection nodes in both conditions. If *plotUnchanged* is FALSE, the unchanged positive selection nodes in both conditions will not be displayed.

```
> biexample=biCkaksAA(example02,example)
> result=biCompare(biexample)

> plot(result)
```



