**S1 Methods. Method for generating *de novo* C-terminal sequence patterns.**

***Steps for generating sequences/patterns***

1. Get a list of protein sequences anchored at the c-termini between 3 and 10 amino acids in length
2. Generate terminal anchored combination supersets of the sequences generated in step 1
3. Eliminate supersets which do not occur more than once

***Step 1: Get a list of protein sequences anchored at the C-termini between 3 and 10 amino acids in length***

Let **Σ** represent the set of amino acids present in proteins of the human genome, then **Σ** can be expressed as:

**Σ = { A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V }**

A single element of the set **Σ** can be expressed as **σp** where **p** represents the amino acid position in **Σ**:

**{ σp ϵ Σ, 1 ≤ p ≤ 20)**

**example: σ19 = Y**

A single element or no element can be symbolized as follows:

**{ σpε | σpε ϵ (Σ U empty set), 0 ≤ p ≤ 20, 0 = empty state}**

**examples: σ3ε= N, σ0ε = { }**

Any generated sequence of length 3 to 10 amino acids can be represented as follows:

**ω = σaσbσcσdεσeεσfεσgεσhεσiεσjε**

**example: σ12σ4σ6σ11σ0εσ0εσ0εσ0εσ0εσ0ε = KDEL**

Let **Ρ** represent the set of proteins in the human genome, then **Ρi** represents a single protein where:

**{ Ρi ϵ Ρ | 1 ≤ Ρi ≤ ΡSIZE }**

A position specific amino acid in protein **Ρi** can be express as follows:

 **Ρi, j = { σp | 1 ≤ j ≤ Ρi, LENGTH }**

 **example: if Ρ1 = {…DMEETDD} then Ρ1, LENGTH-2 = T = σ17**

Let **Ψ** represent the set of C-termini sequences generated from **Ρ**, then **Ψi**  is of the form **ω** where:

 **ΨSIZE = (ΡSIZE \* 8)**

**{ Ψi ϵ Ψ** **| 1 ≤ Ψi ≤ ΨSIZE }**

Generation of the set **Ψ**:

 **for each Ρi ϵ Ρ**

 **j = Ρi, LENGTH – 9**

 **while( j ≤ (Ρi, LENGTH – 2) ) do the following:**

 **ω = Ρi, j + ... + Ρi, LENGTH-1 + Ρi, LENGTH**

**add ω to Ψ**

 **j = j + 1**

**example:**

**Ρi** = {…GNMGPQYVTTYA}

**Ψ =** {MGPQYVTTYA, GPQYVTTYA, PQYVTTYA, QYVTTYA, YVTTYA, VTTYA, TTYA, TYA}

***Step 2: Generate terminal anchored combination supersets of the sequences generated in step 1***

A single degenerate position can be denoted by **Χ**, which represents all amino acids:

**Χ = { σp ϵ Σ }**

A single degenerate position or no element at that position can be denoted as **Χε**:

 **Χε = { σp ϵ (Σ U empty set) }**

A sequence with at least one degenerate position whereby neither the first nor the last position can be degenerate can be expressed as follows:

**φ = { σa(σxε Χε)^\*Χ (σxε Χε)^\*σb | ^\* symbolizes the shorthand expansion of (σxε Χε) }**

Let **Φ** represent the set of degenerate sequences derived from **Ψ**, then **Φi** is a degenerate sequence of the form φwhere:

 **ΦSIZE =** $\sum\_{i=1}^{ΨSIZE}\sum\_{r=1}^{Ψi, LENGTH-2}\frac{\left(Ψi, LENGTH-2\right)!}{\left(Ψi, LENGTH-2-r\right)!\*r!}$

**{ Φi ϵ Φ** **| 1 ≤ Φi ≤ Φsize }**

Generation of the set **Φ**:

For each **Ψi**

j=1

 while ( j < 2**Ψi, LENGTH-2**) do:

 φ **= Ψi, 1**

 mod\_bin = j

 k=2

 while (k < **Ψi, LENGTH**) do:

 if the remainder of mod\_bin/2 is 1, do:

 φ **=** φ **+ Χ**

otherwise:

 φ **=** φ **+ Ψi, k**

mod\_bin = mod\_bin/2

 k = k + 1

 φ **=** φ **+ Ψi, LENGTH**

 add φ **to Φ**

 j = j + 1

***algorithm b.***

**example:**

**Ψ** = { MGPQYVTTYA, GPQYVTTYA, PQYVTTYA, QYVTTYA, YVTTYA, VTTYA, TTYA, TYA }

**Φ =** { MXPQYVTTYA , MGXQYVTTYA , MXXQYVTTYA , MGPXYVTTYA, …, TXYA, TTXA, TXXA, TXA}

***Step 3: Eliminate supersets which do not occur more than once***

Let **Ω** represent a set of all degenerate and non-degenerate sequences derived from **Ψ** and **Φ**, then **Ωi** is either of the form **ω** or **φ** where:

**{ Ωi ϵ Ω | 1 ≤ Ωi ≤ Ω SIZE }**

Generation of the set **Ω**:

for each **Ψj**:

 add **Ψj** to **Ω**

for each **Φj**:

 count = 0

 for each **Ψk:**

if **Ψk**  matches **Φj** -> (note: all **Χ** in **Φj** can be substituted for any amino acid at the same position in **Ψk**)

 count = count + 1

 if count > 1

 add **Φj** to **Ω**

***algorithm c.***

**example:**

**Ψ = {** …, TGYA, ATYA, TTTA, TTRA, … **}**

**Φ = {** …, TXYA, TTXA, TXXA, …}

**Ω = {** …, TGYA, ATYA, TTTA, TTRA, …, TTXA, TXXA, … }