

Genesis.Generator

The generator produces a set of feature extractor or filter functions $[f_1, f_2, \dots, f_N]$, where each function f_i is defined as:

$$f_i : \langle Sequence \rangle \rightarrow \langle Vector \text{ of fixed length } P \rangle$$

For each protein sequence in the database, these functions are evaluated to compute tables T_1, T_2, \dots, T_N .

Each table T_i corresponds to a sequence S_i , and is structured as a symbol table:

Symbol	Value
f_1	$\langle Vector \rangle$
f_2	$\langle Vector \rangle$
...	...
f_N	$\langle Vector \rangle$

Thus, every sequence S_i in the database has an associated table T_i containing the outputs of all functions f_1 through f_N .

Functions f_1, \dots, f_k , where k is a fixed positive integer, are **predefined and hard-coded**. These serve as foundational features and are not dynamically generated. The first three of these predefined functions are:

$$\text{feature_hydrophobicity}, \quad \text{feature_molecular_weight}, \quad \text{feature_secondary_structure}$$

Functions f_{k+1}, \dots, f_N , on the other hand, are **dynamically generated**. Each of these functions is defined as a **polynomial combination** (e.g., weighted sum, non-linear composition) of one or more **preceding functions**. For instance, f_{k+1} may depend on f_2 and f_5 , and be defined as:

$$f_{k+1}(x) = 0.4 \cdot f_2(x) + 0.6 \cdot f_5(x)$$

for $k > 5$

These dynamic functions are randomly generated at initialization or through an optimization process (e.g., evolutionary algorithms or PSO). While f_{k+1} to f_N may have dependencies on any earlier functions, the predefined functions f_1 to f_k are independent and act as **building blocks** for generating the dynamic functions.