

### **Traccia 1:**

Descrivere la procedura operativa per il calcolo di un modello QSAR (*Quantitative Structure-Activity Relationship*) per la stima del fattore di bioconcentrazione (BCF) di sostanze organiche. Descrivere, anche con l'ausilio del *dataset*-esempio bcf.xlsx messo a disposizione, le diverse operazioni necessarie per la calibrazione e validazione di un modello di regressione, con particolare attenzione agli aspetti di predittività, complessità, applicabilità, interpretabilità degli effetti variabili, presenza di dati anomali.

Il fattore di bioconcentrazione (BCF) è il rapporto tra la concentrazione della sostanza chimica nel pesce e la sua concentrazione allo stato stazionario nel mezzo circostante. Questo parametro viene usato per individuare gli inquinanti ambientali in grado di accumularsi negli organismi acquatici. La normativa europea (REACH) prevede la seguente classificazione delle sostanze sulla base del valore del BCF:

- se  $\log(\text{BCF}) > 3.3$  la sostanza chimica è etichettata come bioaccumulabile
- se  $\log(\text{BCF}) < 3.3$  la sostanza chimica è etichettata come non-biocumulabile

Il *dataset*-esempio (bcf.xlsx) include 473 sostanze organiche, elencate nelle righe di colonna A mediante uno specifico codice alfanumerico, di cui 81 bioaccumulabili e 392 non-bioaccumulabili (colonna H). Per ciascuna sostanza, sono stati calcolati i cinque descrittori molecolari (MLOGP, AEige, SpMax2\_Bh(p), GATS5v, CI-089), i cui valori numerici sono riportati nelle colonne B – F. La colonna G contiene i valori in scala logaritmica del fattore di bioconcentrazione (LOGBCF).

### **Prova di inglese:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

## Traccia 2:

Descrivere la procedura operativa per il calcolo di un modello QSAR (*Quantitative Structure-Activity Relationship*) per la classificazione di sostanze organiche in bioaccumulabili e non-bioaccumulabili in funzione del loro fattore di bioconcentrazione (BCF). Descrivere, anche con l'ausilio del *dataset*-esempio bcf.xlsx messo a disposizione, le diverse operazioni necessarie per la valutazione preliminare del grado di sovrapposizione delle classi nello spazio chimico selezionato e per la successiva calibrazione/validazione di un modello di classificazione, con particolare attenzione agli aspetti di predittività, complessità e applicabilità del modello a nuove sostanze.

Il fattore di bioconcentrazione (BCF) è il rapporto tra la concentrazione della sostanza chimica nel pesce e la sua concentrazione allo stato stazionario nel mezzo circostante. Questo parametro viene usato per individuare gli inquinanti ambientali in grado di accumularsi negli organismi acquatici. La normativa europea (REACH) prevede la seguente classificazione delle sostanze sulla base del valore del BCF:

- se  $\log(\text{BCF}) > 3.3$  la sostanza chimica è etichettata come bioaccumulabile
- se  $\log(\text{BCF}) < 3.3$  la sostanza chimica è etichettata come non-bioaccumulabile

Il *dataset*-esempio (bcf.xlsx) include 473 sostanze organiche, elencate nelle righe di colonna A mediante uno specifico codice alfanumerico, di cui 81 bioaccumulabili e 392 non-bioaccumulabili (colonna H). Per ciascuna sostanza, sono stati calcolati i cinque descrittori molecolari (MLOGP, AEige, SpMax2\_Bh(p), GATS5v, CI-089), i cui valori numerici sono riportati nelle colonne B – F. La colonna G contiene i valori in scala logaritmica del fattore di bioconcentrazione (LOGBCF).

## Prova di inglese:

Several C# features help create robust and durable applications. **Garbage collection** automatically reclaims memory occupied by unreachable unused objects. **Nullable types** guard against variables that don't refer to allocated objects. **Exception handling** provides a structured and extensible approach to error detection and recovery. **Lambda expressions** support functional programming techniques. **Language Integrated Query (LINQ)** syntax creates a common pattern for working with data from any source. Language support for **asynchronous operations** provides syntax for building distributed systems. C# has a **unified type system**. All C# types, including primitive types such as int and double, inherit from a single root object type. All types share a set of common operations. Values of any type can be stored, transported, and operated upon in a consistent manner. Furthermore, C# supports both user-defined reference types and value types. C# allows dynamic allocation of objects and in-line storage of lightweight structures. C# supports generic methods and types, which provide increased type safety and performance. C# provides iterators, which enable implementers of collection classes to define custom behaviors for client code.