Pharmaceutical Bioinformatics Primer

Ola Spjuth

September 10, 2017

Contents

1 Sequence analysis

The central dogma in molecular biology describes how the genetic information that we inherit from our parents is stored in deoxyribonucleic acid (DNA), and how this information is used to make identical copies of DNA, and how this DNA is converted to RNA and to protein. DNA is a linear polymer of the four bases adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T); for RNA, the base uracil (U) is present instead of thymine. Proteins are polypeptides made up of sequentially arranged amino acids forming a chain and these chains folds into distinct structures. In biology, structure determines function, and it is the 3D structure of the proteins that determines their functions, such as the protein's ability to catalyze a reaction, metabolize a drug, or transport a molecule over a cell-membranes. A *gene* is a sequence of DNA which can be transcribed into mRNA for further translation into a protein.

1.1 Representation of sequences in computers

Due to the fact that the information in DNA and RNA is contained in the order of the bases, such sequences are commonly represented in computers by their primary structure in a simple text string. Proteins can also be represented by their primary structure.

1.1.1 FASTA format

The by far most commonly used file format for biological sequences is FASTA (see Figure 1 for an example). The format stores nucleic acid or protein sequences as character strings in a plain text format. Sequences in FASTA formatted files are preceded by a line starting with the character '>'. The first word on this line is commonly the ID or name of the sequence, and the rest of the line a description of the sequence. The remaining lines contain the sequence itself. Blank lines are ignored, and so are spaces and other gap symbols (dashes, underscores, periods) in a sequence. The FASTA format supports multiple sequences by adding more sequences to the end of the file. A common way of annotating FASTA entries is to separate them with the character '|', see Figure 1.

```
>gi|532319|pir|TVFV2E|TVFV2E envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK
```

Figure 1: An protein sequence in FASTA format with annotations separated by the character '|'.

1.2 Comparing sequences

Comparing two sequences can easily be done by comparing each position in both sequences, and sum up the matches and mismatches. However, in practice sequences are seldom of the same length which complicates the process.

1.2.1 Sequence alignments

A sequence alignment is an arrangement of two or more sequences to each other with the purpose to highlight similarities between the sequences, which can be interpreted as functional, structural, or evolutionary relationships between the sequences. These interpretations are based on the following fundamental assumptions:

1. Sequences that are similar share a common ancestral sequence (homologous).

2. Due to common ancestry, similar sequences have similar functionality.

While it is possible to align simple sequences by hand, longer sequences require computational algorithms for this to be feasible. Global alignments aims at finding the best overall alignment between sequences, "forcing" the alignment to span the full length of all sequences, while local alignments aims at finding short regions of highly conserved sequences.

Since alignments compare sequences the foremost application is to search for similar sequences in databases. Also, an alignment can be used to create a consensus sequence of one or many sequences that summarize a collection of sequences in the best way. Some use cases are to identify sequences in one

species based on what one know in another species, to compare unknown sequences against known ones in order to guess a function, and to find conserved regions between species.

1.2.2 Substitution matrices

It is common for alignments to use a substitution matrix (or scoring matrix) to assign scores to matches or mismatches, and a gap-penalty for matching a residue in one sequence to a gap in the other. The substitution matrix describes the rate at which one character in a sequence changes to other characters over time, and gives a measure of how common a mutation from *e.g.* one amino acid is to another amino acid. Higher scores indicate that the probability that those two amino acids aligned by chance is very small. Lower scores indicate a high probability that the two amino acids are aligned by chance, and that they are thus evolutionarily unrelated. It is common that matches which are unlikely to have been a result of evolution are given negative scores. The two most commonly used types of scoring matrices are the PAM matrices and the BLOSUM matrices.

1.3 BLAST

BLAST (Basic Local Alignment Search Tool) is an algorithm and tool used to find local sequence alignments between proteins and nucleotide sequences. For molecular biology, it has become the de-facto tool for searching for similar sequences in a database (*"please BLAST this sequence against GenBank"*).

1.4 Example applications of sequence analysis

1.4.1 Detecting Open Reading Frames

The fact that DNA nucleotides need to be read three at a time to specify a protein sequence implies that a DNA sequence has three different reading frames, which are determined by whether we start at nucleotide one, two, or three. Since both strands of DNA can be copied into RNA, a DNA sequence with its complementary strand can specify six different reading frames. An *open reading frame* (ORF) is a portion of a sequence that could potentially encode a protein, and the start- and end points are not equal to the mRNA but usually contained within the mRNA's sequence. In a gene, ORFs are

Alignment:

```
G A T T A C A T T A T A
G A T G - - A T A A C -
```

Substitution matrix:

```
      A    C    G     T
A     8    3    1     3
C          7    3     1
G               6     3
T                     8
```

```
gap open: -3
gap extension: -1
```

The score of the alignment is calculated like this:

```
G  A  T  T  A  C  A  T  T  A  T  A
G  A  T  G  -  -  A  T  A  A  C  -

6  8  8  3  -3 -1 8  8  3  8  1  -3  = 46
```

Figure 2: Example of the scoring of a sequence alignment. We start from left to right and score each position in the sequences individually by looking up the score in the substitution matrix. When a gap of length ¿1 is reached, the first gap position is assigned the score -3 ('gap open') and subsequence gap positions are assigned the score -1 ('gap extension'). We then sum up the scores for all positions to get the score of the alignment.

located between the start codon and the stop codon. It is a common bioinformatics task to identify ORFs from a plain sequence.

1.4.2 SNP detection

The genome differs between individuals, giving us different phenotypes such as length, color, but it also affect how we develop diseases and respond to pathogens, chemicals, drugs, vaccines, and other agents. A *single-nucleotide polymorphism* (SNP) is a DNA sequence variation occurring when a single nucleotide - A, T, C, or G - in the genome differs between members of a species or paired chromosomes in an individual. This means that one fraction

of the population may have one nucleotide and another fraction may have another one. An example is the gene BRCA1, located on chromosome 17, which has a SNP where a certain base is T for 68 % of the population, and C for 32 % of the population. This polymorphism is of considerable interest as it has been shown to affect the risk of developing breast- and ovarian cancer. Identifying SNPs and relate them to disease traits and other factors of biology is thus an important bioinformatics task, and is commonly carried out by comparison of genomes for large populations.

2 Cheminformatics

2.1 Chemical graphs

While many chemists prefer to work with images of chemical structures, computers do not deal well with chemical drawings and therefore needs to use a different representation. For cheminformatics the *chemical graph* is the most prominent underlying representation, and this representation is used for most of the chem- and bioinformatics involved in pharmaceutical studies. The advantages and limitations of graph-based representations are important to know, as the limitations are an important source of error in statistical modeling. For example, chemical graphs are static and do not describe well important chemical properties such as delocalization and tautomerism.

For molecules, we can represent the atoms and bonds as vertices connected by edges. Each atom is a *vertex* and each bond is an *edge*. This representation is quite close to the way we draw molecules as 2D diagrams, namely as atoms with lines in between them to indicate the bonding. However, this only defines how the atoms are connected. To represent a molecular structure, we need to add additional information: for each atom we would like to specify the element (by its symbol), and for each bond we want to specify if it is a single, double, or triple bond.

The primary use of a chemical graph is as a mathematical model of a molecule. As such, we can use it in computers as a data model, as an exchange format, and to compare molecular structures.

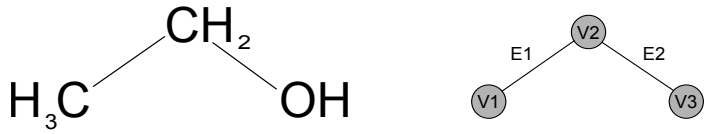


Figure 3: Ethanol and its graph. The atoms of the molecule maps to the vertices V1, V2, and V3 of the graph; the two bonds map to the edges E1, and E2. The carbon and oxygen elemental symbols are colors of the vertices, while the single bond orders color the edges.

2.2 Connection tables

A *connection table* is an alternative representation of the information in a chemical graph where each atom and bond is explicitly given as separate facts. Most chemical file formats make use of some connection table representation. A connection table lists the vertices of the chemical graph and their properties. A simple example where the vertices are labeled by element symbol and bonds by formal order could look like this:

```
# Atoms
C
O
C
# Bonds
1 2 1
1 3 2
```

The table thus includes two sub-tables: one for atoms, and one for bonds. The atom table has the element symbol, the bond table has three columns beginning with the two atom numbers which define the linked atoms by index; the last column has the order information for the bond. For example atom 1 is linked to atom 2 by a single bond, and atom 1 is linked to atom 3 with a double bond.

The most common chemical file format that stores molecules as a connection table is MDL Mol-files. An example Mol-file and its connection table is shown in Figure 4. One drawback of the MDL molfile format is that it does not support additional data connected to the chemical structure. The chemical format SDF (structure-data file) wraps a molfile and adds support for associated data. Further, SDF files can have multiple compounds separated by four dollar signs ($$$$). The data properties of one chemical structure in the SDF file can look like the following (following directly after the molfile):

```
>  <LogS (exp)>
-2.4300001

>  <EXTREG>
1

>  <ClogP>
```

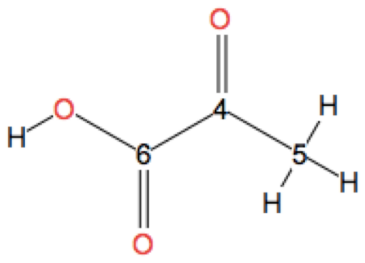
```
1.54

>  <Entry>
10024055

>  <Polarizability>
223.5
```

```
  Mrv0541 09201116592D

 10  9  0  0  0  0            999 V2000
    2.0930    0.2063    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    3.5220    1.0313    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    2.8075   -1.0313    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    3.5220    0.2063    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    4.2365   -0.2063    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.8075   -0.2063    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.9807   -0.6492    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
    4.6794   -0.4620    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
    4.4922    0.2367    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
    1.6500   -0.0495    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
  1  6  1  0  0  0  0
  1 10  1  0  0  0  0
  2  4  2  0  0  0  0
  3  6  2  0  0  0  0
  4  5  1  0  0  0  0
  4  6  1  0  0  0  0
  5  7  1  0  0  0  0
  5  8  1  0  0  0  0
  5  9  1  0  0  0  0
```



Pyruvic acid

Figure 4: The mol-file of pyruvic acid showing the connection table.

2.3 SMILES

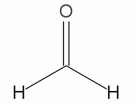
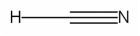
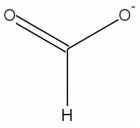
Simplified Molecular Input Line Entry Specification, or SMILES, is a line notation for molecular structures that provides a simple syntax where chemical structures are described as a simple one-line text string that contains each non-hydrogen atom in the notation with their elemental symbol.

2.3.1 SMILES: Atoms and bonds

SMILES denote atoms by their element symbol wrapped in two square brackets. For example, silicon is denoted [Si]. However, for the elements C, N, O, S, and the halogens the brackets can be omitted that reduces the size of SMILES for organic compounds considerably. However, the organic subset notation cannot be used for an atom if additional information, such as formal charge, is defined for that atom. Therefore, a charged oxygen is denoted as [O-], and a positively charged nitrogen as [N+].

Bond orders are indicated with '-', '=' and '#'. Disconnected atoms are separated by a '.', which for example can be used to represent salts. When no explicit bond order is specified, usually a single bond is assumed. A few example SMILES strings are given in Table 1.

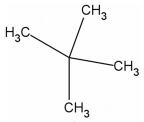
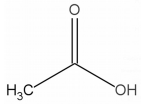
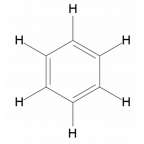
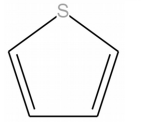
Table 1: Some examples of SMILES showing how atoms, bonds, and formal atomic charges are represented.

SMILES	**Molecular Formula**	**Name**	**Molecular Structure**
C	CH_4	methane	
CCO	C_2H_6O	ethanol	
C=O	CH_2O	formaldehyde	
C#N	CHN	hydrogen cyanide	
O=C[O-]	$[CHO_2]^-$	formate	

2.3.2 SMILES: Branches and cycles

To accommodate branching parenthesis are used. For example, isobutane is represented as *CC(C)C*. Cycles are created by defining links numerically, by numbering the two atoms that close the cycle. A digit 1-9 is appended to the atom to indicate that it is involved in a ring closure. For example, cyclohexane can be represented with the SMILES C1CCCCC1. Table 2 shows a few examples involving branches and cycles.

Table 2: Some examples of SMILES showing how branches and cycles are represented.

SMILES	**Molecular Formula**	**Name**	**Molecular Structure**
CC(C)(C)C	CH_5H_12	neopentane	
CC(=O)O	$C_2H_4O_2$	acetic acid	
C1=CC=CC=C1	C_6H_6	benzene	
S1C=CC=C1	C_4H_4S	thiophene	

2.4 SMARTS

SMARTS is a query language that extends SMILES and provides means to represent substructure searches. For example, if we would like to express that some atom can be an oxygen or a nitrogen, we can express this in a SMARTS query. Table 3 gives a subset of additional expressions that SMARTS provides on top of SMILES. A more in-depth document is available at `http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html`.

Table 3: SMARTS extends SMILES with additional query functionality. Some examples are given here to broaden the scope of the query.

	SMARTS	Description
Atoms	*	Any atom.
	a	Any aromatic atom.
	A	Any aliphatic atom.
	[Ca]	A calcium atom.
	[O,N]	An oxygen or a nitrogen.
	[R]	A ring atom.
Bonds	-	A single bond.
	=	A double bond.
	#	A triple bond.
	:	An aromatic bond.
	~	Any bond.
	@	A ring bond.

2.5 Comparing chemical structures

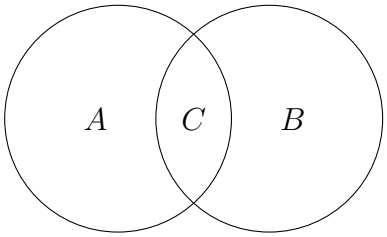
Comparing chemical structures is an important task in order to e.g. search for similar molecules in a database, often assuming the "similarity principle" that similar structures have similar properties.

2.5.1 Molecular fingerprints

Like human fingerprints are distinct for the person they belong to, *molecular fingerprints* are distinct for the molecule they represent: each chemical graph has a distinct fingerprint by which one can recognize the molecule. This chemical fingerprint typically looks like a string of numbers, and the numbers are often bits (which can have values 0 or 1) where each bit indicates the presence or absence of particular chemical features, such as the presence or absence of a particular substructure (see Figure 5 for an example). The molecular fingerprint is not necessarily unique, as many molecules can, and often do, have the same fingerprint.

A prominent application of the fingerprint is *database searching*, where

```
struct A:       00010100010101000101010011110100    13 bits on  (A)
struct B:       00000000100101001001000011100000    8 bits on   (B)
A AND B:        00000000000101000001000011100000    6 bits on   (C)
```



$$tanimoto = \frac{C}{A + B - C} = \frac{6}{13 + 8 - 6} = 0.4$$

Figure 5: Tanimoto distance between two binary fingerprints describing two chemical structures. C is the number of bits set in both fingerprints, and A and B are the number of bits set in each fingerprint individually.

one tries to find information about chemical structures in a database. Matching two molecules by chemical graphs is a computationally relatively expensive operation. By contrast, comparing two fingerprints is much faster. Therefore, a database look-up is normally implemented by first matching fingerprints. Because the fingerprint is not unique the matching may lead to a number of false hits. The fingerprint matching is therefore commonly followed by a chemical graph matching.

The most easily interpretable fingerprint is the substructure fingerprint. Each bit maps here one-to-one to an exact chemical substructure. A substructure can be a phenyl ring, a carboxylic acid, or a amide functional group, for example. If the bit corresponding to a phenyl group has the value 1, then the molecule contains at least one phenyl ring.

2.5.2 Comparing fingerprints

The *Tanimoto similarity* is a common way to compare binary fingerprints, and as such it represents a simple approach to a measure for *molecular similarity*. It calculates the similarity for two bit strings with values between 0 and 1, see Figure 5.

If the structures are identical, then all bits are in common and similarity = 1. If two fingerprints have no bits in common, then similarity = 0. Using this approach, we can define molecular similarity by calculating the Tanimoto similarity based on the fingerprints of the two respective molecules. This approach takes advantage of the fact that if two molecules are very similar, then they will both have the same substructures present in their structures, and therefore the same fingerprint bits are set.

3 Further reading

For further reading, please refer to the book: *Pharmaceutical Bioinformatics* by Wikberg, J. et al. (Oakleaf Academic). Many of the examples in this primer originates from this book.