

UNIX Command Line Interface (CLI)

hotkeys: ↑ or ↓ to scroll e.g. through manual, tab $\xrightarrow{\text{K}}$ for autocompleting file names

| = pipe

\ = quote next character (e.g. space or special characters)

> file = save output in file

directories (absolute paths start with /, relative paths not!)

- . = current (working) directory
- .. = parent directory
- ../.. = grandparent directory
- ~ = home / user directory
- / = root directory

wildcards

- ? = placeholder for any character
- * = several characters (0 ... ∞)
- [sequence] = regular expression (e.g. [A-F] or [5-9] or ["ATCG"])

permissions

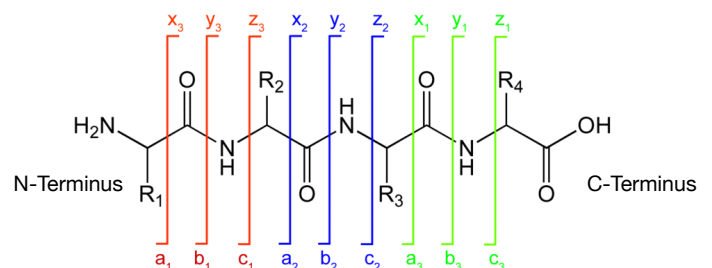
- Read determines if the content of a directory is visible
- Write determines if new files can be created or deleted
- Execute determines if a user can change (cd) into the directory

command	description	details / options
ls	list files of working directory	-a (all), -l (details), -h (human readable file sizes)
history	history of used commands	
man	manual page of command	
pwd	print working directory	
cd	change working directory	
cp	copy files	
mv	move/rename files	
touch	create empty file	
mkdir	create empty directory	
rm	remove files	
rmdir	remove empty directory	
cat	concatenate content of files	
less	page through contents of files	
wc	word count	-l (print number of lines), -w (print number of words)

command	description	details / options
grep	search patterns	<text> <filename> -v (print lines NOT containing the text), -l (print only names of files containing the text), -n (print line numbers as well)
wget	get files from WWW (Linux)	
tar	archive files	zxf <archive> (extracts contents of compressed archive, file indicated by .tar.gz or .tgz), zcf <archive> <file1> [<file2>, ...] (creates compressed archive and adds file1 and file2 to it), ztf <archive> (displays archive content without extracting it)
top	list programs/processes	
ps	process select (by ID)	-f (details)
kill	kill program/process	
df	disk free space	-a (including dummy file systems, -h (human readable), -l (only local file systems)
du	disk usage	-a (all files, not directories), -c (total), -h (human readable), --max-depth=N (maximal depth of N)
chown	change owner	[option] [owner] [:[group]] file -R (operate on files recursively), -v (output diagnostic for every file processed)
chmod	change mode (permissions)	u (owner user), g (owner group), o (others), a (all users) e.g. chmod g+w file

Peptidfragmentierung

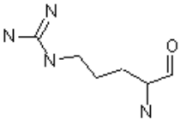
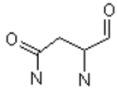
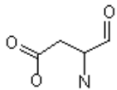
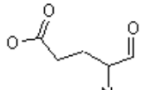
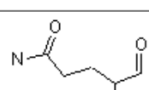
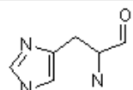
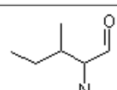
Peptidfragmente (meist b- oder y-Ionen) werden bei der Tandem-Massenspektroskopie nach m/z (Masse pro Ladung) angeordnet. Ziel ist es, die ursprüngliche Sequenzabfolge der Aminosäuren herauszufinden, indem man die Peaks des Spektrums (m/z-Werte) den passenden Aminosäuren zuordnet. Der Einfachheit halber wird in BILS meist angenommen, dass z=1 ist, d.h. man muss nur die Massen beachten, nicht die Ladung.

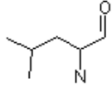
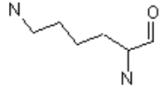
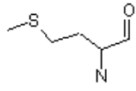
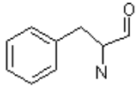
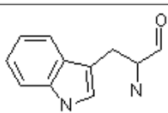
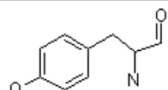


Spektrums (m/z-Werte) den passenden Aminosäuren zuordnet. Der Einfachheit halber wird in BILS meist angenommen, dass z=1 ist, d.h. man muss nur die Massen beachten, nicht die Ladung.

- b-Ionen: Aminosäuren ausgehend vom N-terminalen Ende des Peptids
Restmasse (engl. residue mass) einer Aminosäure = Masse wenn Aminosäure im Peptid gebunden ist, d.h. von der typischen Masse in der Tabelle muss 1 Da abgezogen werden (für das fehlende H-Atom)
- y-Ionen: Aminosäuren ausgehend vom C-Terminus des Peptids
Restmasse = typische Masse der AS + 19 Da (für +H₂O +H)
- beim ersten y-Ion (y₁) handelt es sich um dieselbe AS wie beim letzten b-Ion (b_n), y₂=b_{n-1}, y₃=b_{n-2} etc., allerdings sind natürlich die m/z-Werte verschieden
- Trypsin schneidet ein Protein nur an Stellen mit den Aminosäuren Lysin oder Arginin in Peptide, d.h. tryptische Peptide (engl. tryptic peptides) enden immer mit Lysin oder Arginin. Erwartete charakteristische Masse für das y₁-Ion: entweder m/z=128+19=147 für Lysin oder m/z=156+19=175 für Arginin (für z=1)

- Zuerst muss also das y_1 -Ion ermittelt werden. Anschließend addiert man die typische Masse einer beliebigen Aminosäure zum m/z -Wert von y_1 (z.B. +71 für Alanin). Falls das Ergebnis mit einem Peak aus dem Spektrum übereinstimmt, haben wir die Aminosäure des y_2 -Ions identifiziert. Fortfahren (Masse einer beliebigen Aminosäure zum m/z -Wert von y_2 addieren usw.) bis alle Aminosäuren bekannt sind.
- Man kann natürlich auch vom b_1 -Ion ausgehen, falls man durch logisches Überlegen herausfindet, welcher Wert dies sein könnte: z.B. bei einem m/z -Wert von 88 ($88=87-1$) muss das b_1 -Ion Serin sein. Auch hier addiert man die typische Masse einer beliebigen Aminosäure, bis man ein mit dem Spektrum übereinstimmendes Ergebnis erhält.

Residue	3-letter code	1-letter code	Mono-isotopic mass	Average mass	Structure
Alanine C_3H_5NO	Ala	A	71.037114	71.0779	
Arginine $C_6H_{12}N_4O$	Arg	R	156.101111	156.1857	
Asparagine $C_4H_6N_2O_2$	Asn	N	114.042927	114.1026	
Aspartic acid $C_4H_5NO_3$	Asp	D	115.026943	115.0874	
Asn or Asp	Asx	B			
Cysteine C_3H_5NOS	Cys	C	103.009185	103.1429	
Glutamic acid $C_5H_7NO_3$	Glu	E	129.042593	129.114	
Glutamine $C_5H_8N_2O_2$	Gln	Q	128.058578	128.1292	
Glu or Gln	Glx	Z			
Glycine C_2H_3NO	Gly	G	57.021464	57.0513	
Histidine $C_6H_7N_3O$	His	H	137.058912	137.1393	
Isoleucine $C_6H_{11}NO$	Ile	I	113.084064	113.1576	

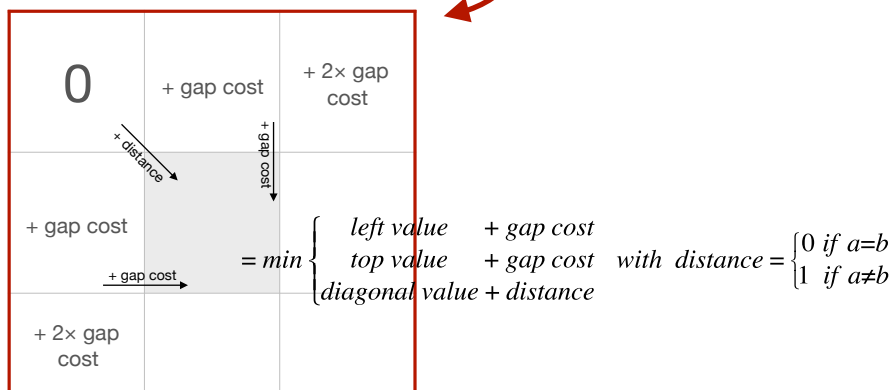
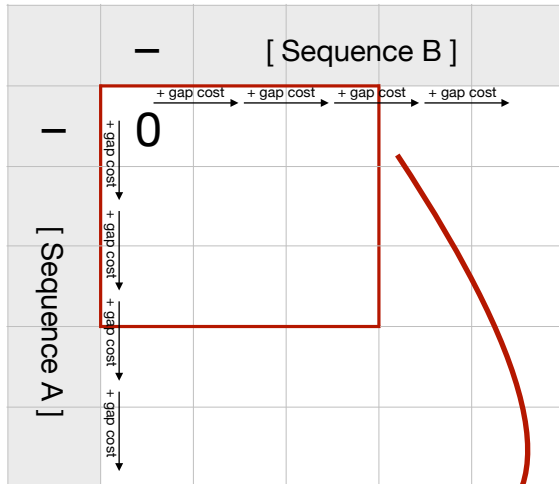
Residue	3-letter code	1-letter code	Mono-isotopic mass	Average mass	Structure
Leucine $C_6H_{11}NO$	Leu	L	113.084064	113.1576	
Lysine $C_6H_{12}N_2O$	Lys	K	128.094963	128.1723	
Methionine C_5H_9NOS	Met	M	131.040485	131.1961	
Phenylalanine C_9H_9NO	Phe	F	147.068414	147.1739	
Proline C_5H_7NO	Pro	P	97.052764	97.1152	
Serine $C_3H_5NO_2$	Ser	S	87.032028	87.0773	
Threonine $C_4H_7NO_2$	Thr	T	101.047679	101.1039	
Selenocysteine C_3H_5NOSe	Sec	U	150.95363	150.0379	
Tryptophan $C_{11}H_{10}N_2O$	Trp	W	186.079313	186.2099	
Tyrosine $C_9H_9NO_2$	Tyr	Y	163.06332	163.1733	
Unknown	Xaa	X			
Valine C_5H_9NO	Val	V	99.068414	99.1311	

Needleman-Wunsch-Algorithmus

Dynamic programming for pairwise alignment

1. Scoring-Matrix

- Sequenz A und Sequenz B in die Kopfzeile und Spalte (nach dem Platzhalter –) eintragen
- Zahl 0 in erste Zelle einfügen, sowie in der ersten Zeile und Spalte immer die gap cost/gap penalty addieren (bei gap cost = 2 wäre die erste Zeile 0, 2, 4, 6, 8 usw.)



- neue Zelle wird als das Minimum aller drei Möglichkeiten definiert (falls die gap cost subtrahiert wurde oder negativ ist → neue Zelle ist das Maximum aller Möglichkeiten). Gewählte Wege mit Pfeilen markieren (ein Weg oder mehrere mit gleichen Werten möglich).
- distance(a,b) bezieht sich auf die aktuelle Aminosäure aus Sequenz A und B. Hier wird die einfache Distanzfunktion gewählt (der Wertebereich wird vielfältiger, wenn stattdessen eine Ähnlichkeitsfunktion s(a,b) verwendet wird, die z.B. auf der chemischen oder evolutionären Austauschbarkeit basiert).
- Gesamte Matrix berechnen

2. Traceback Matrix

- Die Score-Matrix wird in eine Traceback-Matrix überführt (dieser Schritt ist nicht unbedingt notwendig und kann weggelassen werden, da bereits durch die Pfeilmarkierungen klar ist, welcher Weg bei der Berechnung das Minimum war). Die Werte für eine Zelle ergeben sich folgendermaßen: 0 wenn der Wert von diagonal kam, -1 wenn der Wert von oben kam, 1 wenn man von links kam.

3. Alignment

- Man startet in der unteren rechten Ecke und geht den eingeschlagenen Weg entlang der Pfeile oder Traceback-Werte zurück, bis man auf die 0 trifft. Alle möglichen Wege stellen eine optimale Anordnung dar.
- Die Anordnung beginnt vom Ende aus, d.h. man schreibt die Sequenzen von rechts nach links. Wenn man von diagonal kam, schreibt man die Aminosäuren beider Sequenzen untereinander. Falls man von oben kam, erhält Sequenz B einen Platzhalter –, für Sequenz A schreibt man die entsprechende Aminosäure hin. Kam man von links, bekommt Sequenz A einen Platzhalter: –.

Smith-Waterman-Algorithmus

Hier wird die Scoring-Matrix anders berechnet! Die gap cost ist für gewöhnlich negativ → man berechnet die neue Zelle über das Maximum (statt Minimum). Die Null im Wertebereich macht lokale Alignments von Teilsequenzen möglich (bei globalem Alignment der gesamten Sequenz fällt die Null wieder aus dem Wertebereich weg!). Globales Alignment macht nur dann Sinn, wenn man eine hohe Ähnlichkeit beider Sequenzen erwartet.

$$= \max \begin{cases} 0 \\ \text{top value} - \text{gap cost} \\ \text{left value} - \text{gap cost} \\ \text{diagon. value} + s(a,b) \end{cases}$$

$s(a,b)$ = similarity Funktion, z.B. BLOSUM oder PAM (Ähnlichkeit ist Aminosäuren-spezifisch definiert)

Beispiel: BLOSUM50 Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	J	Z	X	*
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-2	-1	-1	-5
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3	-1	-3	0	-1	-5
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3	5	-4	0	-1	-5
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4	6	-4	1	-1	-5
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	-3	-2	-3	-1	-5
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3	0	-3	4	-1	-5
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3	1	-3	5	-1	-5
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4	-1	-4	-2	-1	-5
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4	0	-3	0	-1	-5
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4	-4	4	-3	-1	-5
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1	-4	4	-3	-1	-5
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3	0	-3	1	-1	-5
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1	-3	2	-1	-1	-5
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1	-4	1	-4	-1	-5
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3	-2	-3	-1	-1	-5
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2	0	-3	0	-1	-5
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0	0	-1	-1	-1	-5
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3	-5	-2	-2	-1	-5
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1	-3	-1	-2	-1	-5
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	-3	2	-3	-1	-5
B	-2	-1	5	6	-3	0	1	-1	0	-4	-4	0	-3	-4	-2	0	0	-5	-3	-3	6	-4	1	-1	-5
J	-2	-3	-4	-4	-2	-3	-3	-4	-3	4	4	-3	2	1	-3	-3	-1	-2	-1	2	-4	4	-3	-1	-5
Z	-1	0	0	1	-3	4	5	-2	0	-3	-3	1	-1	-4	-1	0	-1	-2	-2	-3	1	-3	5	-1	-5
X	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-5
*	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1