Bioinformatyka 1

Wprowadzenie i sprawy organizacyjne. BASH i wyrażenia regularne.

Krzysztof Murzyn

Zakład Biofizyki Obliczeniowej i Bioinformatyki
Wydział Biochemii, Biofizyki i Biotechnologii
Uniwersytet Jagielloński

Plan wykładu

1. **Bioinformatyka**
   - Definicje
   - Zastosowania
2. **Podstawowe informacje o kursie**
   - Terminy zajęć i warunki zaliczenia
   - Zawartość kursu
   - Biblioteka użytkownika kursu
   - Cele kursu
   - Praktyczne wskazówki
3. **GNU/Linux**
   - System operacyjny i system plików
   - BASH: powłoka tekstowa
   - Wyrażenia regularne

Kontakt

dr hab. Krzysztof Murzyn
Zakład Biofizyki Obliczeniowej i Bioinformatyki
WBBiB UJ, pok. B028 (wtorek 11:00-11:45)
tel. (12) 664-63-79
email: krzysztof.murzyn@uj.edu.pl, Teams Chat
http://bioinfo.mol.uj.edu.pl/modmol/People/KrzysztofMurzyn

Bioinformatyka

nowa dziedzina nauki rozwijająca się na styku biologii molekularnej, w której rozmaite zgadnienia biologiczne rozważane są w kategoriach cząsteczek i fizyko-chemicznych oddziaływań między nimi, oraz informatyki, stosującej narzędzia matematyczne i metody obliczeniowe w celu:

- zgromadzenia,
- sklasyfikowania,
- masowego przetwarzania i
- zrozumienia

informacji związanej z biocząsteczkami

Informacja

sekwencja nukleotydowa : ACGT/U, gen $\approx$ 1000 nukleotydów (1 Kn), ponad 40 kompletnych genomów (0.6 Mn $\div$ 3.2 Gn $\div$ 0.12 Tn), transkryptom (jednoczesny pomiar transkrypcji ok. 6000 genów).

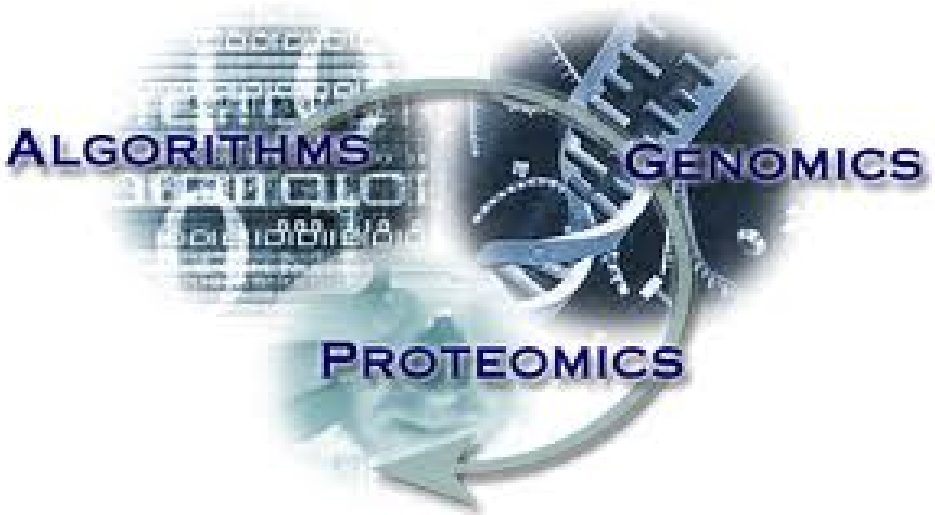
sekwencja aminokwasowa : 20 rodzajów reszt aminokwasowych, średniej wielkości białko $\approx$ 300 reszt, domena $\approx$ 200 reszt; znanych sekwencji – ponad 300 000, domen – ponad 10 000, zwojów (ang. *fold*) – ponad 13 000.

struktura przestrzenna makrocząsteczek : znanych struktur – ponad 13 000, średnio po 1 000 atomów.

literatura naukowa : $\approx 10^1$ mln cytowań rocznie.

metadane : *dane o danych*; szlaki metaboliczne (KEGG Pathways), ontologie (hierarchicznie uporządkowany system nazewnictwa, np. Gene Ontology)

Informatyka



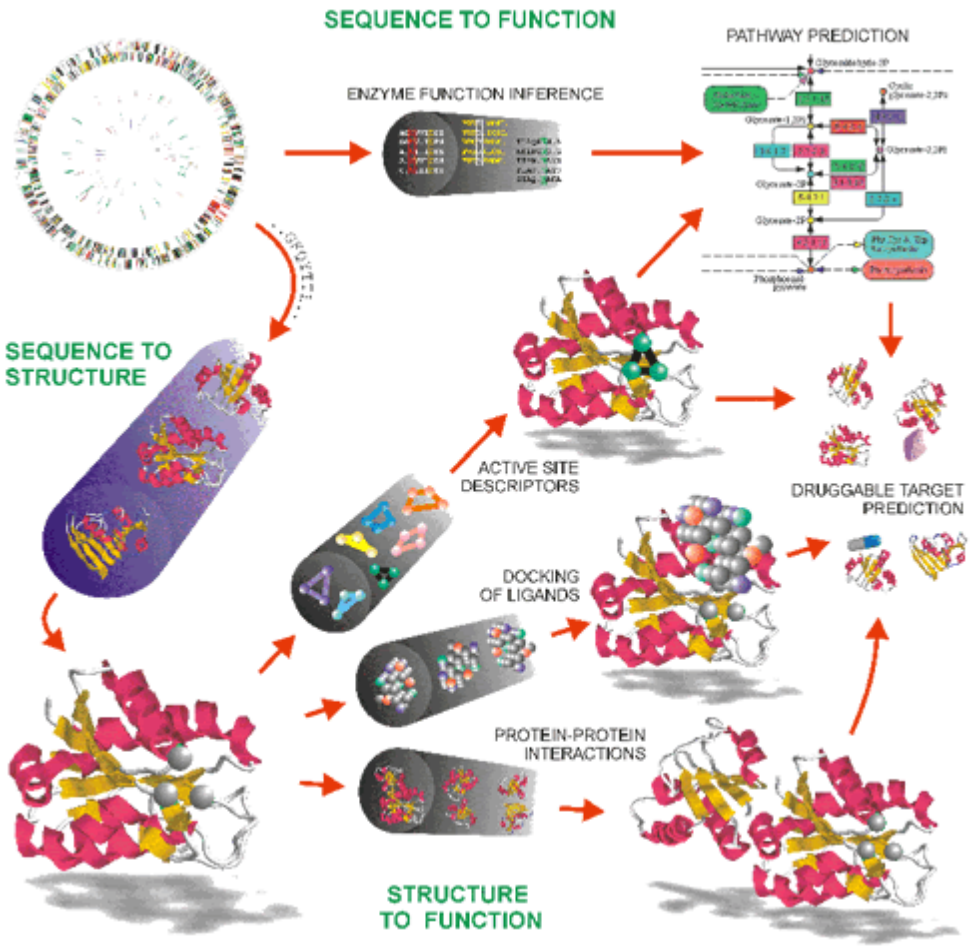
Informatyka dotyczy w takim samym stopniu komputerów jak astronomia teleskopów.

narzędzia relacyjne bazy danych (SQL), języki programowania (C, C++, Fortran, Python, Perl, Java), systemy operacyjne (Unix, Windows), programy użytkowe (edytory tekstu, grafiki, przeglądarki internetowe), protokoły transmisji (HTTP, FTP, SSH), standardy prezentacji (HTML, XHTML, PDF), przechowywania i wymiany informacji (XML)

metody programowanie dynamiczne, algorytmy heurystyczne, sieci neuronowe, algorytmy genetyczne, modelowanie probabilistyczne (HMM), geometria komputerowa, analiza ciągów tekstowych, wyrażenia regularne

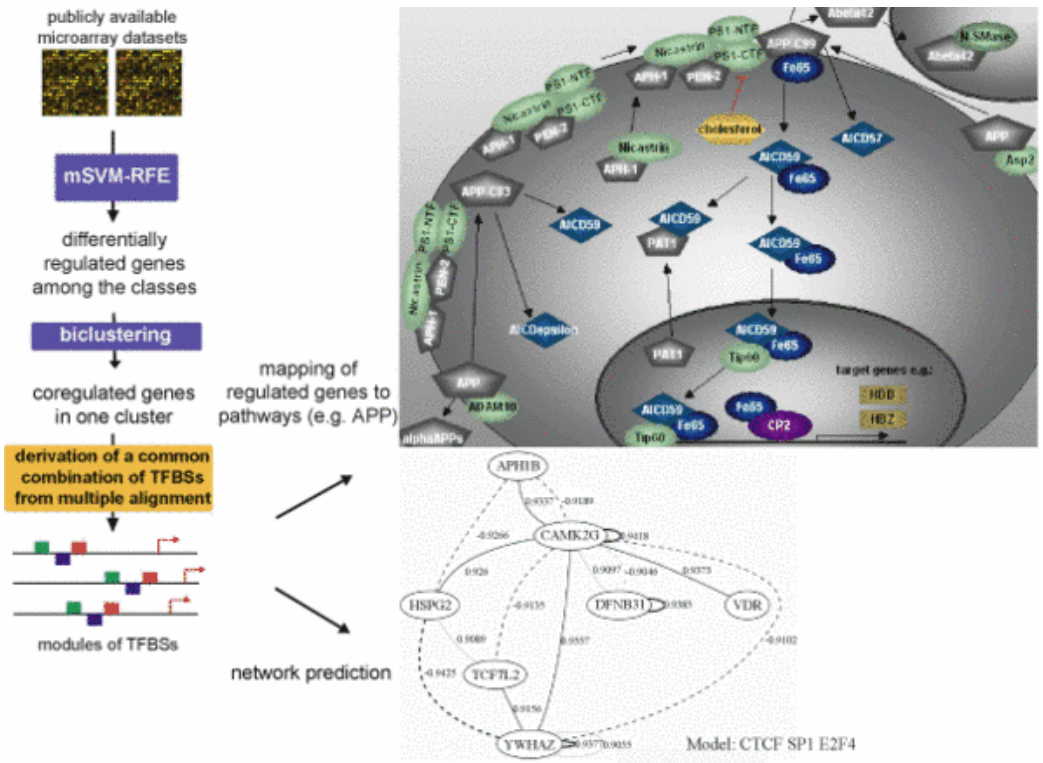
Obszary badań

- gromadzenie i przetwarzanie różnorodnych danych (sekwencje aminokwasowe i nukleotydowe, literatura naukowa, mikromacierze, widma masowe, etc.), konstrukcja systemów eksperckich (diagnostyka medyczna etc.)
- przewidywanie struktury i funkcji białkowych i niebiałkowych produktów ekspresji genów,
- dopasowywanie sekwencji (ang. *alignment*), identyfikacja motywów w strukturze pierwszorzędowej, przewidywanie funkcji nieznanych białek, badanie ewolucji molekularnej



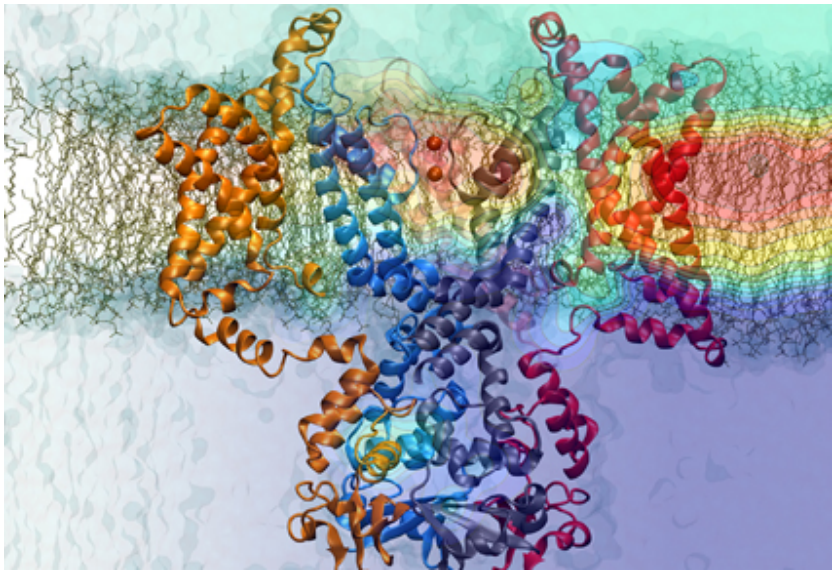
Obszary badań

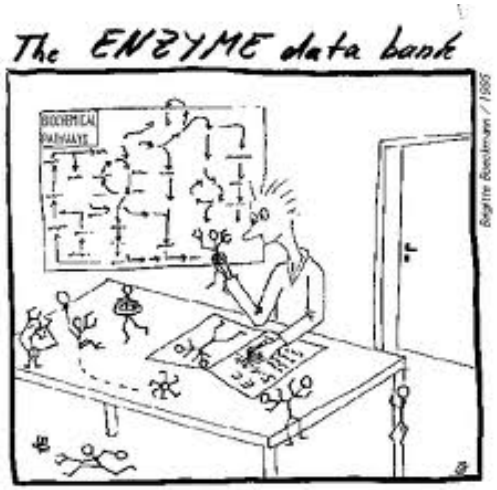
MASOWE PRZETWARZANIE INFORMACJI NA POZIOMIE GENOMÓW



- analiza struktury genów
- charakterystyka sekwencji promotorowych kontrola ekspresji, identyfikacja miejsc wiązania czynników transkrypcyjnych, charakterystyka położenia i roli sekwencji rozproszonych w genomach,
- analiza strukturalno-funkcjonalna genomu genomika porównawcza,
- mikromacierze korelacja profilów ekspresji genów w zróżnicowanych warunkach

Obszary badań





- modelowanie molekularne (badania podstawowe nad funkcjonowaniem i własnościami błon lipidowych, związkami między strukturą i dynamiką a funkcją białek, etc; wspomagane komputerowo projektowanie leków (CADD)), analiza geometryczna (powierzchnia, objętość),
- analiza kontroli metabolicznej (MCA, ang. Metabolic Control Analysis)
- informatyzacja omików (proteomika, genomika, metabolika, lipidomika...) – rozwój programów użytkowych wspomagających i ułatwiających prowadzenie rutynowych analiz związanych z pracę laboratoryjną (np. ExPaSy Proteomics Server, http://www.expasy.org)

Zajęcia w ramach kursu i warunki zaliczenia

- 10 wykładów (łącznie 20h), środa 8:30-10, sala D107, stacjonarny test pojedynczego wyboru $(+2/-1/0)$ z pytaniami otwartymi i zamkniętymi
- 12 ćwiczeń (po 3h, 135 min), stacjonarny test praktyczny (zestaw zadań do samodzielnego rozwiązania) - łącznie 40h, wszystkie ćwiczenia stacjonarnie
- ćwiczenia rozpoczynają się w tygodniu następującym po pierwszym wykładzie, możliwości odrabiania opuszczonych zajęć bardzo ograniczone, każdorazowo za zgodą ćwiczeniowca o ile liczba osób w grupie zajęciowej nie przekroczy 14 osób
- oceny cząstkowe (testy, ćwiczenia) w systemie USOSWeb (moduł Sprawdziany)

- zaliczenie kursu obejmuje:
  - zaliczenie ćwiczeń (ZAL 50+/NZAL)
    wykonanie ćwiczeń $(2 + 3.5 \cdot 10$ pkt), wynik próbnego testu na ostatnich ćwiczeniach $(13$ pkt), wynik testu praktycznego (120 min, ok. 8 zadań, 50 pkt); łącznie: 100 pkt
  - zaliczenie wykładu – ocena do średniej
    warunek wstępny: pozytywna ocena z ćwiczeń, wynik testu pojedynczego wyboru zawierającego pytania z zagadnień poruszanych na wykładach i ćwiczeniach (teoria, maks. 100 pkt) oraz punktowy wynik zaliczenia ćwiczeń (maks. 100 pkt), łącznie: 200 pkt

Skala ocen	
bdb	≥ 90 %
+db	$[\,80, 90\,)$ %
db	$[\,70, 80\,)$ %
+dst	$[\,60, 70\,)$ %
dst	$[\,50, 60\,)$ %
ndst	< 50 %

Wykłady

1. Wprowadzenie i sprawy organizacyjne. BASH i wyrażenia regularne.
2. Internetowe zasoby informacji o genach i białkach. Bazy danych: PROSITE, PRINTS, UniProt. Serwisy: NCBI Entrez i Web of Knowledge.
3. RCSB PDB i wizualizacja struktury makrocząsteczek (PyMOL).
4. Elementarna analiza sekwencji. Pakiet EMBOSS. Macierze kropkowe.
5. Macierze punktacji różnicą logarytmiczną. Dopasowania pary sekwencji.
6. Algorytmy heurystyczne: FASTA i BLAST
7. Dopasowanie wielu sekwencji (MSA): metody wyznaczania, edycji, formaty i porównania. Baza danych dopasowań referencyjnych BaliBase.
8. Architektura domenowa białek. Profilowe ukryte modele Markowa. Bazy danych PFAM, SCOP, CATH.
9. Molekularna analiza filogenetyczna - modele ewolucji sekwencji
10. Molekularna analiza filogenetyczna - algorytmy konstrukcji drzew filogenetycznych

---

Prezentacje z wykładów (PDF) będą na bieżąco udostępniane w materiałach kursowych w dedykowanym zespole Teams **Bioinformatyka 1 [WBT-BINF1.7] 2022**

Ćwiczenia 1 – 6

1. powłoka systemu GNU/Linux BASH. Praca w Win10 (WSL, SublimeText, WinSCP, PowerShell, PyMOL, ClustalX), praca zdalna: ssh/scp/sftp (wzgl. WSL: emboss, phylip), wyrażenia regularne (grep).
2. wizualizacja struktury makrocząsteczek w PyMOL; pomiary kątów płaskich, torsyjnych, odległości; porównywanie struktur 3d
3. praca z danymi literaturowymi (NCBI PubMed, Web of Knowledge)
4. bazy danych PROSITE i PRINTS, serwis InterPro - motywy sekwencyjne, motywy strukturalne, wykorzystanie śladów sekwencyjnych w identyfikacji i klasyfikacji białek błonowych oraz białek zaw. fragmenty o niskiej złożoności składu
5. wprowadzenie do EMBOSS. Macierze kropkowe - białka mozaikowe, powtórzenia motywów, fragmenty o niskiej złożoności składu
6. dopasowania sekwencji (globalne, lokalne); macierze punktacji różnicą log.; ocena dopasowania nominalna vs bitowa

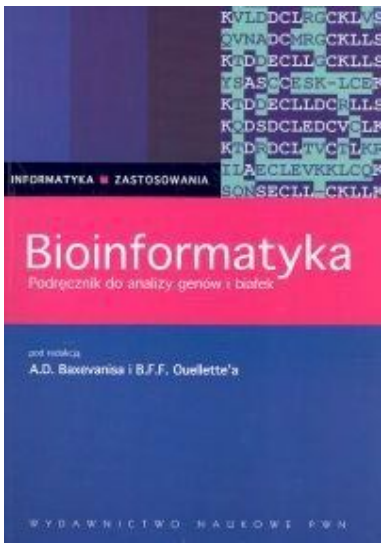
---

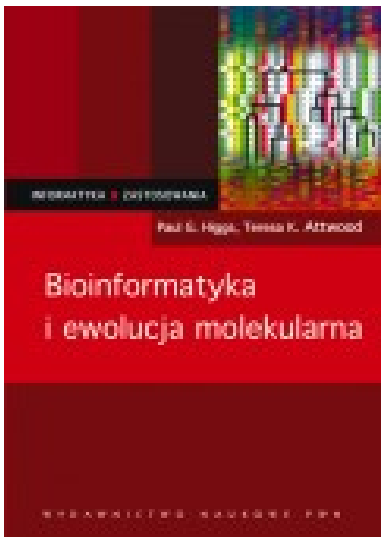
- materiały dotyczące ćwiczeń udostępniane przez ćwiczeniowca

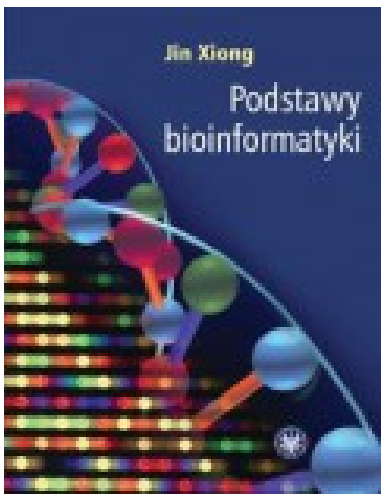
Ćwiczenia 7 – 12

7. algorytmy heurystyczne: FASTA, prss, fastf lub fasts - studia przypadków z odpowiednich publikacji (W. Pearson)
8. algorytmy heurystyczne: BLAST (PSI-BLAST) - wszystkie homologi w PSI-BLAST (weryfikacja trafień z SwissProt i klasyfikacji w PROSITE), rozmywanie profilu w PSI-BLAST, BLAST nucleotide i Bl2seq
9. program ClustalX, baza danych BaliBase, edycja i konwersja MSA
10. molekularna analiza filogenetyczna - pakiet Phylip, modele ewolucji sekwencji, test MLR
11. molekularna analiza filogenetyczna - metody konstrukcji drzew, bootstrap
12. test zderzeniowy: oswajenie się z komputerami w pracowniach WBBiB, praca indywidualna, dyskusja nad wybranymi zagadnieniami dot. przygotowania do testu praktycznego

Interesujące książki







- Bioinformatyka i ewolucja molekularna, T Attwood & P Higgs ****
- Podstawy bioinformatyki, J Xiong **
- Bioinformatyka. Podręcznik do analizy genów i białek., AD Baxevanis & BFF Outellette, *
- Wprowadzenie do bioinformatyki., A Lesk, ***

Wiedza i umiejętności

Wiedza

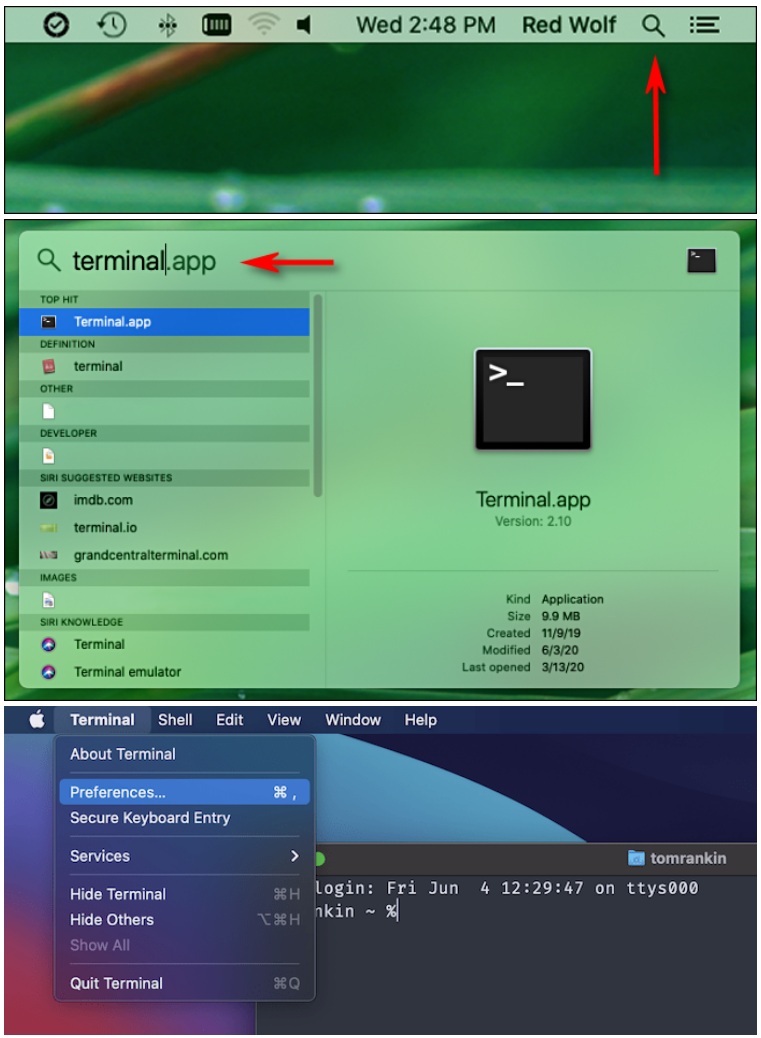
- wyszukiwanie i identyfikacja białkowych i nukleotydowych sekwencji homologicznych
- konstrukcja dopasowań sekwencji
- jakościowe porównywanie sekwencji metodą macierzy kropkowych
- wyszukiwanie informacji w rekordach wybranych baz danych
- interpretacja i ocena wiarygodności wyników przeszukiwania baz danych sekwencji białek/genów
- wizualizacja struktury przestrzennej makrocząsteczek
- analiza danych literaturowych (Pubmed, Web of Science)
- założenia i przebieg molekularnej analizy filogenetycznej
- analiza domenowej architektury białek

Umiejętności

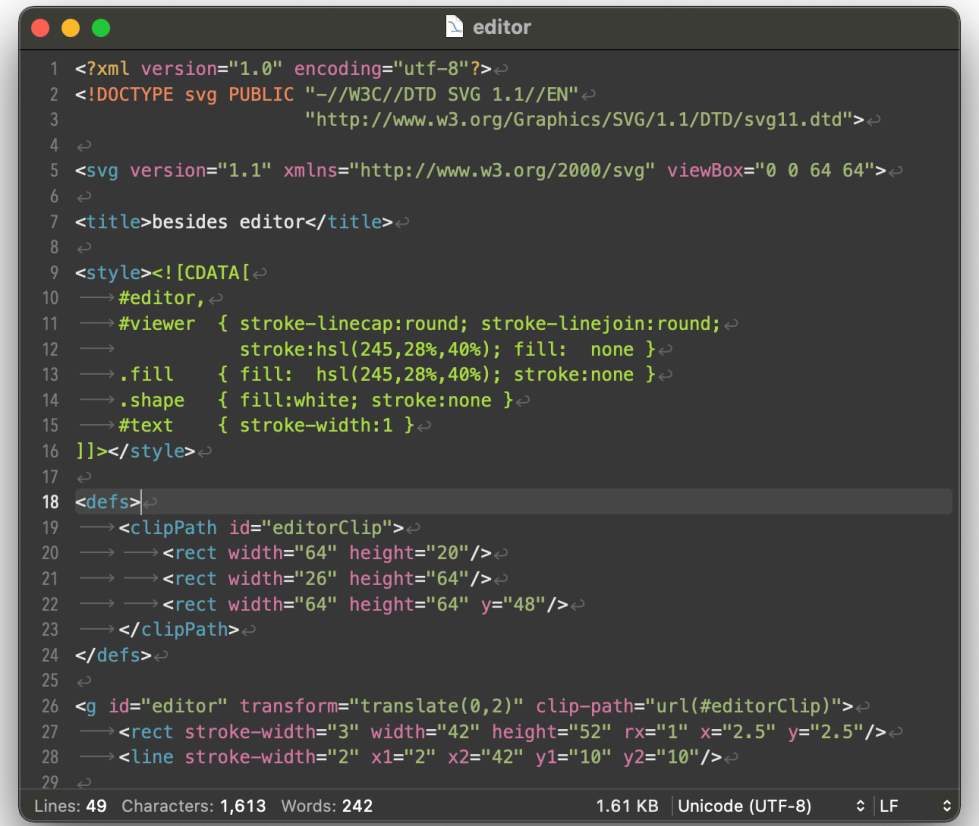
- obsługa specjalistycznego oprogramowania bioinformatycznego: EMBOSS, PyMOL, BLAST, FASTA, Phylip, ClustalX itd.
- korzystanie ze specjalistycznych serwisów internetowych: NCBI Entrez, WebOfScience i baz danych: RCSB PDB, EMBL, UniProt, PubMed
- praca w powłoce tekstowej BASH systemu GNU/Linux

MacOSX

Terminal tekstowy



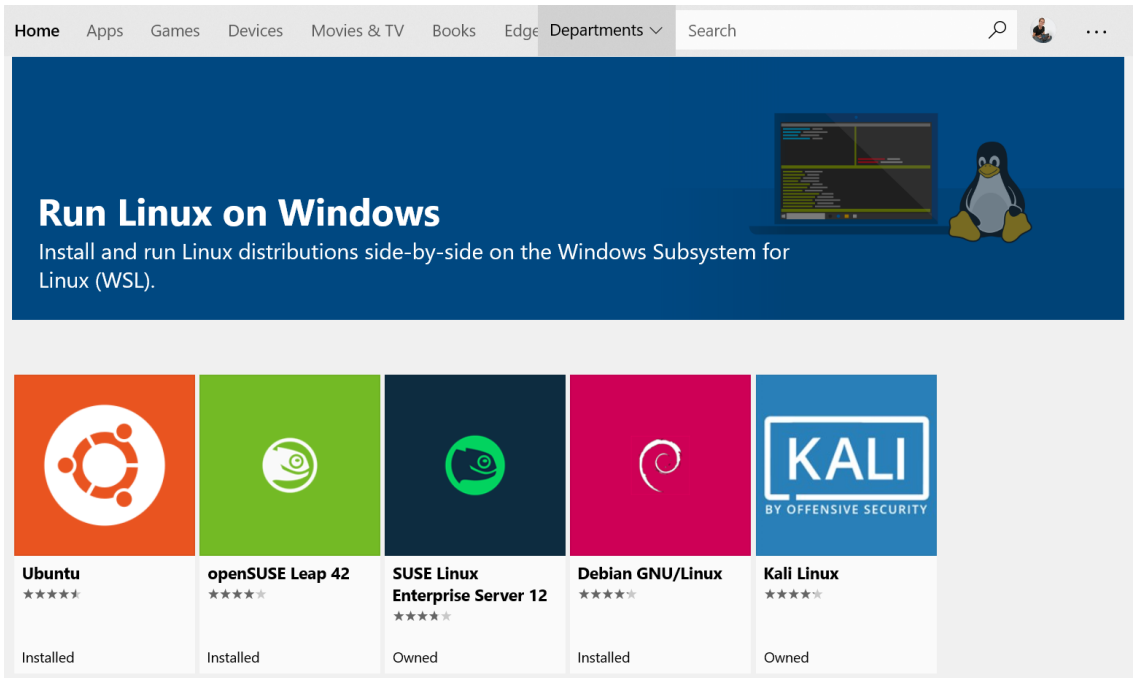
- Edytor tekstu (np. CotEditor, coteditor.com)



- Zmiana domyślnego programu powłoki (zsh) na coś innego (np. bash, csh): chsh -s /bin/bash.

Windows 10+

- Windows Subsystem for Linux, (bash, ssh, apt install emboss)
  instalacja: https://docs.microsoft.com/en-us/windows/wsl/install,
  dobre praktyki: https://docs.microsoft.com/en-us/windows/wsl/setup/environment



- edytor tekstu (VSCode (dla bardziej zaawansowanych), SublimeText (lekki i minimalistyczny)) i manager plików z obsługą ssh (WinSCP)
- rekomendowany emulator terminala tekstowego: Windows Terminal (Microsoft Store)

MacOSX/Win10+/Linux: CONDA

- Conda otwartoźródłowy systemem zarządzania pakietami oprogramowania dostępny na conda.io dla systemów Win10+, Linux i MacOSX napisany w języku Python
- Conda umożliwia tworzenie izolowanych środowisk programistycznych (dowolny język programowania) oraz instalowanie/aktualizowanie/usuwanie pakietów oprogramowania z uwzględnieniem zależności między nimi
- dystrybucja pakietów realizowana jest w kanałach (ang. channels), w których zgrupowano (zazw. tematycznie) różnorodne oprogramowanie
- minimalistyczną wersją systemu Conda jest Miniconda (przy instalacji wybiera się domyślną wersję interpretera Pythona, np. Python v3.7); wersje interpretera można później łatwo zmienić (upgrade/downgrade)
- wyszukiwanie pakietów oprogramowania można zrealizować z linii poleceń (np. conda search pymol) lub on-line na stronie anaconda.org (Anaconda to Conda poszerzona o zestaw pakietów Pythona (głównie z zakresu Danetyki (Data Science), wzgl. Inżynierii i Analizy Danych)

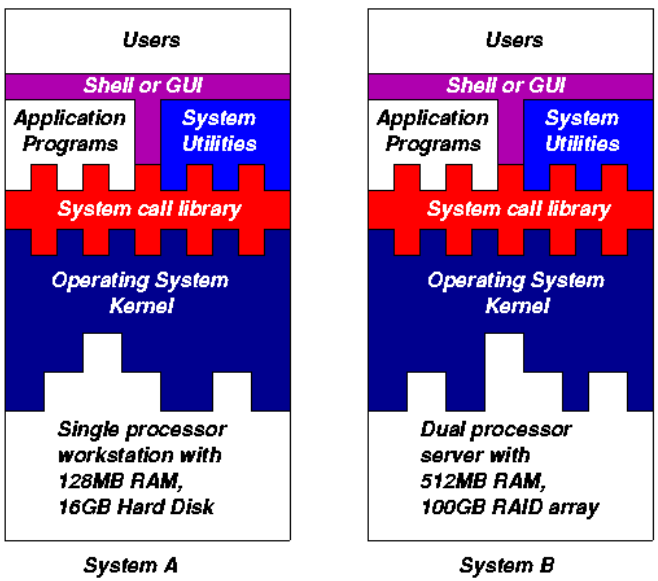
System operacyjny

System operacyjny (SO) jest oprogramowaniem, którego poszczególne części (procedury) umożliwiają użytkownikom (ludzie, programy użytkowe) wykorzystywanie sprzętowych zasobów komputera w bezpieczny, wydajny i abstrakcyjny sposób

bezpiecznie oznacza np., że tylko jeden program użytkowy może w danej chwili przesyłać dane bezpośrednio na drukarkę

wydajnie oznacza np., że programy, które oczekują na przesłanie tych danych wykorzystują czas procesora w najmniejszym możliwym zakresie

abstrakcyjnie oznacza np., że użytkownik operuje plikami a nie fizyczną lokalizacją danych na dysku dzięki czemu do pracy z komputerem nie jest konieczna znajomość parametrów technicznych jego podzespołów

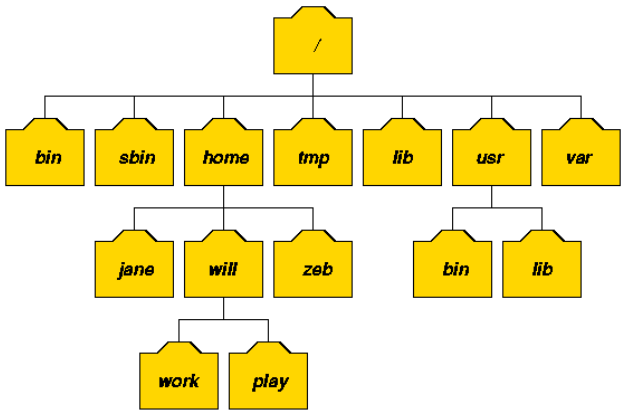


SO oferuje użytkownikom (programom użytkowym) ujednolicony dostęp do komputera niezależnie od parametrów technicznych jego podzespołów. Główne składniki SO:

- jądro i wywołania systemowe
- system plików
- powłoka (graficzna / tekstowa)

System plików

- każdy plik jest najmniejszym przydziałem logicznej pamięci pomocniczej, w którym zapisana jest informacja (dane tekstowe lub binarne, kod wykonywalny programu)
- nazwy plików to niemal dowolny ciąg maksymalnie 256 znaków
- nazwa nie może zawierać znaku /
- ze względów praktycznych zaleca się nie używanie znaku odstępu oraz znaków specjalnych (np. `* ? # & > < |`, nawiasy oraz znaki diakrytryczne `Ł ą Ę ć`...)



hierarchiczna struktura drzewa katalogowego

zwykły plik umożliwia przechowywanie danych (tekstowych i binarnych) lub kod wykonywalny samodzielnego programu lub biblioteki programistycznej

kartoteka umożliwia grupowanie elementów systemu plików

powiązania wskaźniki do innych elementów systemu plików

sztywne dowiązanie (ang. *hard link*) jest nieodróżnialne od pliku, na który wskazuje

symboliczne dowiązanie (ang. *symbolic link*) jest skrótem do danego pliku i przechowywany jest jako element kartoteki ze ścieżką opisującą lokalizację docelowego pliku w drzewie katalogowym

Lokalizacja plików w systemie plików

Aby określić położenie wybranego pliku w systemie, należy podać ścieżkę dostępu zdefiniowaną w oparciu o topologię drzewa katalogowego

Ścieżka bezwzględna

- opisuje zawsze jeden i ten sam ściśle określony plik bez względu na to, w którym miejscu drzewa katalogowego użytkownik się znajduje.

Ścieżka względna

- określa położenie danego pliku względem bieżącej lokalizacji użytkownika na drzewie katalogowym; jeden plik może mieć kilka względnych ścieżek.
- Pliki o takiej samej nazwie znajdujące się w różnych miejsca drzewa katalogowego mogą posiadać takie same ścieżki względne

`/home/murzyn/dydaktyka/pytania_do_egzaminu.odt`

`../bioinfo-1.2021/czwartek15/pymol`
`~/dydaktyka/bioinfo-1.2021/czwaretek15/pymol`

- **UWAGA**: UNIX rozróżnia duże i małe litery (ang. casesensitive); trzeba o tym pamiętać wpisując nazwę konta i hasło (niewyświetlane na ekranie): Sy7koNg $\neq$ Sy7kong

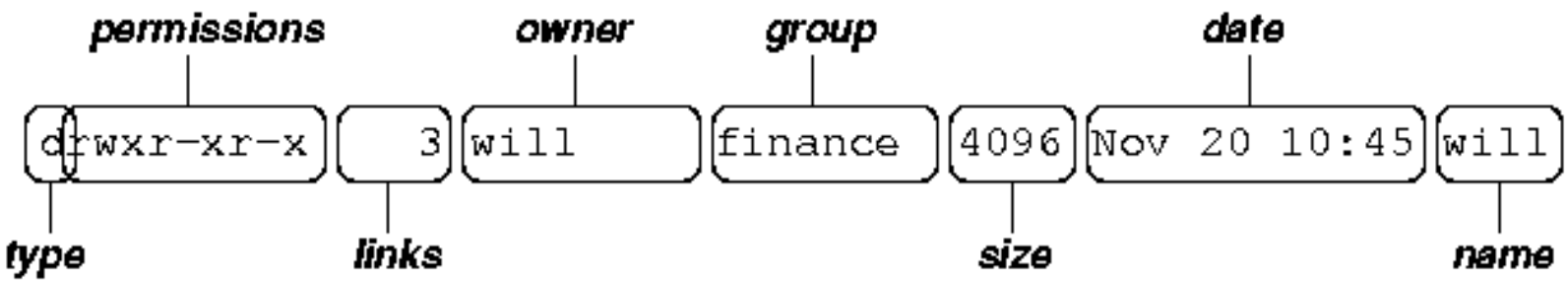
BASH - filozofia korzystania z powłoki tekstowej

Kluczowe dla obsługi systemu GNU/Linux i najczęściej wykorzystywane programy narzędziowe i użytkowe mają kilka wspólnych cech:

- zwarta składnia poleceń - krótkie nazwy, wykorzystanie możliwie szerokiego zakresu znaków ASCII
- dane wyjściowe - preferowane dane tekstowe, dzięki czemu można je przetwarzać z wykorzystaniem innych programów ogólnego zastosowania
- każdy program zazwyczaj ma jedno, ściśle określone zastosowanie
- programy można uruchamiać sekwencyjnie i dzięki temu przekazywać dane wyjściowe z jednego programu jako dane wejściowe do kolejnego w zadanym potoku, powłoka tekstowa pozwala na tworzeni skryptów w celu realizacji bardziej złożonych zadań
- prefereowane wykorzystanie istniejących prostych programów/narzędzi zamiast tworzenia nowych i bardziej złożonych

Listowanie, uprawnienia, prawa własności

```
$ ls -l TopSecret
Total 40
drwx------ 2 stud-HK biotech   520 Sep 12 11:23 sprawozdania/
-r-x------ 1 stud-HK biotech 19512 Aug 30 17:45 wyniki.txt
lrwxrwxrwx 1 stud-HK biotech     9 Aug 23 12:22 kosz -> /dev/null
```



typ – pojedynczy znak:

- `d` kartoteka
- `-` zwykły plik
- `l` skrót
- `b` plik specjalny reprezentujący urządzenie blokowe
- `c` plik specjalny reprezentujący urządzenie znakowe

liczba twardych powiązań – w przypadku katalogu oznacza liczbę podkatalogów (min. 2, tj. `.` oraz `..`)

uprawnienia – 3 znaki określające kolejno uprawnienia przyznane użytkownikowi, grupie do której on należy oraz pozostałym użytkownikom (łącznie 9 znaków):

- `r` można odczytać zawartość pliku
- `w` można zmodyfikować jego zawartość
- `x` plik może być uruchamiany (w przypadku kartoteki pozwala na dostęp do jej elementów)
- `--` brak odpowiednich uprawnień

Życie poza powłoką

```
$ chown -R arni:circus California
$ chmod u=rwx,g+rx,g-w,o=--- votes
$ ls -l votes
-rwxr-x--- 1 arni circus     0 Oct  9 18:02 votes
```

pomoc	dokumentacja (help, apropos, info)
katalogi	tworzenie, usuwanie (mkdir, rmdir) oraz wyświetlanie i przeszukiwanie (ls, find)
pliki	kopiowanie, przenoszenie/zmiana nazwy, usuwanie, tworzenie powiązań między plikami (cp, mv, rm, ln), łączenie i dzielenie (cat, dd, head, tail), zmiana praw dostępu i własności (chmod, chown)
pamięć masowa	zajętość przestrzeni dysku i systemów plikowych (du, df)
procesy	wyświetlenie statusu (ps, top)
edytory tekstu	liniowe (ed), strumieniowe (sed), ekranowe (vi, emacs, pico)
tekst	wyświetlanie (more, less) i przetwarzanie: filtrowanie, porównywanie, poprawianie, sortowanie (grep, diff, patch, sort)
programowanie	języki interpretowane (awk, php, python), kompilatory (gcc, g77)

Przekierowania i przetwarzanie potokowe

cat monday.txt >> diary.txt

powoduje dopisanie zawartości `monday.txt` na końcu pliku `diary.txt`

ls -alR ~ > stan_posiadania.ls-R

powoduje zapisanie na dysku kompletnej zawartości katalogu domowego

history | more

przekierowywuje dane ze `stdout` programu `history` na `stdin` programu `more`; dzięki temu wszystkie operacje wykonane dotychczas w oknie terminala będą wyświetlane strona po stronie

(ls -R / > ~/wszystko.ls-R) >& restricted.log

Bash: `ls -lR >& restricted.log > wszystko.ls-R`

polecenie takie, wydane w środowisku powłoki `tcsh`, pozwoli na utworzenie listy wszystkich plików, do których ma dostęp bieżący użytkownik. Informacja o błędach przy tworzeniu listy, wynikających np. z braku uprawnień do wylistowywania zawartości kartotek, zostaną przekierowane z `stderr` do pliku `restricted.log`

Wyrażenia regularne

ciągi znaków pozwalające tworzyć i opisywać wzorce tekstu. Istnieją wysoce efektywne algorytmy pozwalające wyszukiwać w tekście, ciągi znaków odpowiadające określonemu wzorcowi lub sprawdzać, czy zadany ciąg znaków pasuje do wzorca.

Literały (ang. match-self) reprezentacje określonego ciągu tekstowego

- `c` — tekst jednoznaczny: `AGA` $\rightarrow$ `AGA`, $\neq$ `AGT`
- `\c` — tekst wieloznaczny (znaki specjalne): `GT\.` $\rightarrow$ `GT.` $\neq$ `GTC`

Powielacze (ang. repetition) wielokrotne powtórzenia w ciągach tekstowych

- `?` — co najwyżej jedno wystąpienie (0,1): `AT?G` $\rightarrow$ `AG`, `ATG` $\neq$ `ATTG`
- `*` — dowolna liczba powtórzeń, dopuszczalny brak wystąpienia (0,1,...): `AT*G` $\rightarrow$ `AG`, `ATTTTG`
- `+` — co najmniej jedno wystąpienie, możliwe powtórzenia (1,2,...): `AT+G` $\rightarrow$ `ATG`, `ATTTG`
- `{min,max}` — dopuszczalne od *min* do *max* powtórzeń (*min*,...,*max*): `[ACGT]{2,4}CGCG` $\rightarrow$ `AAACGCG`, `ATCGCG`

Wyrażenia regularne – c.d.

Symbole (ang. match-any) reprezentacje grupy ciągów tekstowych

- `.` dowolny znak z wyjątkiem znaku nowej lini: `GT.` $\rightarrow$ `GTC`, `GTA`
- `[...]` jeden spośród określonych znaków: `[ACTG]` $\rightarrow$ `A`, $\neq$ `U`
- `[^...]` każdy z wyjątkiem wymienionych znaków: `[^CG]` $\rightarrow$ `A`, $\neq$ `C`

Ograniczniki (ang. anchoring) znaczniki początku i końca ciągu tekstowego

- `^` początek: `^AGC.*` $\rightarrow$ `AGCTTAC`, $\neq$ `GCTTAC`;
  negacja vs. początek `^[^CG]TA` $\rightarrow$ `ATA`, $\neq$ `GTA`, $\neq$ `TATA`
- `$` koniec: `.*AGC$` $\rightarrow$ `TTAAGC`, $\neq$ `TTAAC`

Semantyki struktury składniowe

- `...|...` alternatywne wzorce: `AG(T|A)CC` $\rightarrow$ `AGTCC`, $\neq$ `AGGCC`
- `(...)` referencyjne grupowanie wzorców:
  `AGCCGT` $\triangleright$ `(...)(...)` $\triangleright$ `\2\1` $\rightarrow$ `CGTAGC`