BIOL 75302: phytoinformatics

Damon P. Little
The Graduate Center, CUNY & New York Botanical Garden

https://github.com/dpl10/phytoinformatics2023

course syllabus

laboratory exercises (5% each, 60% total)

the two lowest exercise scores will be dropped

take–home final exam (40%)

Assignments are due at the beginning of class on the date specified. **No late assignments will be accepted.**

phytoinformatics

phytoinformatics: bioinformatics with a plant *organismal* biology focus

bioinformatics: the (shotgun) marriage of computers and biology:

- information technology (databases, GIS, etc.)
- phylogenetic s.l. algorithms (trees and derivatives)
- genomics algorithms (sequence manipulation, search, etc.)
- statistics (including population genetics)
- machine learning

truth versus simplification

I will lie to you, by oversimplification, throughout the semester.

I will attempt to give you a basic skill set, but keep in mind this is just the beginning—things get a lot more complicated in real life.

II. *An Argument for Divine Providence, taken from the conſtant Regularity obſerv'd in the Births of both Sexes. By Dr.* John Arbuthnott, *Phyſitian in Ordinary to Her Majeſty, and Fellow of the College of Phyſitians and the Royal Society.*

AMong innumerable Footſteps of Divine Providence to be found in the Works of Nature, there is a very remarkable one to be obſerved in the exact Ballance that is maintained, between the Numbers of Men and Women; for by this means it is provided, that the Species may never fail, nor periſh, ſince every Male may have its Female, and of a proportionable Age. This Equality of Males and Females is not the Effect of Chance but Divine Providence, working for a good End, which I thus demonſtrate:

Let there be a Die of Two ſides, M and F, (which denote Croſs and Pile), now to find all the Chances of any determinate Number of ſuch Dice, let the Binome M+F be raiſed to the Power, whoſe Exponent is the Number of Dice given; the Coefficients of the Terms

Christened.		
Anno.	*Males.*	*Females.*
1629	5218	4683
30	4858	4457
31	4422	4102
32	4994	4590
33	5158	4839
34	5035	4820
35	5106	4928
36	4917	4605
37	4703	4457
38	5359	4952
39	5366	4784
40	5518	5332
41	5470	5200
42	5460	4910
43	4793	4617
44	4107	3997
45	4047	3919
46	3768	3395
47	3796	3536

Christened.		
Anno.	*Males.*	*Females.*
1648	3363	3181
49	3079	2746
50	2890	2722
51	3231	2840
52	3220	2908
53	3196	2959
54	3441	3179
55	3655	3349
56	3668	3382
57	3396	3289
58	3157	3013
59	3209	2781
60	3724	3247
61	4748	4107
62	5216	4803
63	5411	4881
64	6041	5681
65	5114	4858
66	4678	4319

BIOINFORMATICS DISCOVERY NOTE
Vol. 26 no. 23 2010, pages 2933–2935
doi:10.1093/bioinformatics/btq571

Sequence analysis

Advance Access publication October 28, 2010

Statistical distribution of amino acid sequences: a proof of Darwinian evolution

Krystian Eitner[1,2,*], Uwe Koch[3], Tomasz Gawęda[2] and Jędrzej Marciniak[1]

[1]Adam Mickiewicz University, ul. Grunwaldzka 6, 60-780 Poznań, [2]BioInfoBank Institute, Św. Marcin 80/82 lok. 355, 61-809 Poznań, Poland and [3]Lead Dicovery Center, Emil-Figge-Strasse 76a, 44227 Dortmund, Germany

Associate Editor: John Quackenbush

ABSTRACT

Motivation: The article presents results of the listing of the quantity of amino acids, dipeptides and tripeptides for all proteins available in the UNIPROT–TREMBL database and the listing for selected species and enzymes. UNIPROT–TREMBL contains protein sequences associated with computationally generated annotations and large-scale functional characterization. Due to the distinct metabolic pathways of amino acid syntheses and their physicochemical properties, the quantities of subpeptides in proteins vary. We have proved that the distribution of amino acids, dipeptides and tripeptides is statistical which confirms that the evolutionary biodiversity development model is subject to the theory of independent events. It seems interesting that certain short peptide combinations occur relatively rarely or even not at all. First, it confirms the Darwinian theory of evolution and second, it opens up opportunities for designing pharmaceuticals among rarely represented short peptide combinations. Furthermore, an innovative approach to the mass analysis of bioinformatic data is presented.

Contact: eitner@amu.edu.pl

Table 1. Example for searching for three amino acid long sequences (sorting by the number of occurrences)

>Sample Fasta File; ATAATTTAGGATTTAC

Normal search	Offset search
TTT 2	ATA 1
ATT 2	TTT 1
TTA 2	AAT 1
GAT 1	TAG 1
TAG 1	ATT 1
GGA 1	GGA 1
AGG 1	TTA 1
ATA 1	
TAA 1	
AAT 1	
TAC 1	

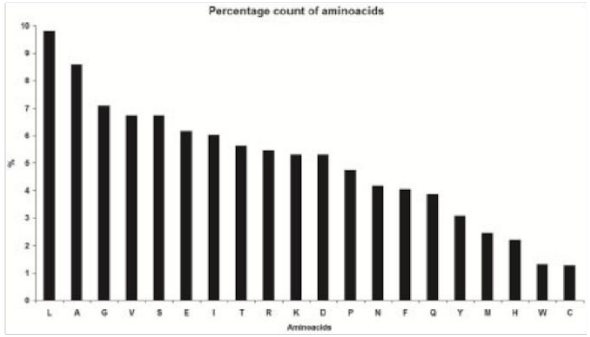


Fig. 1. Amino acid content (%) in the UniProt TREMBL database.

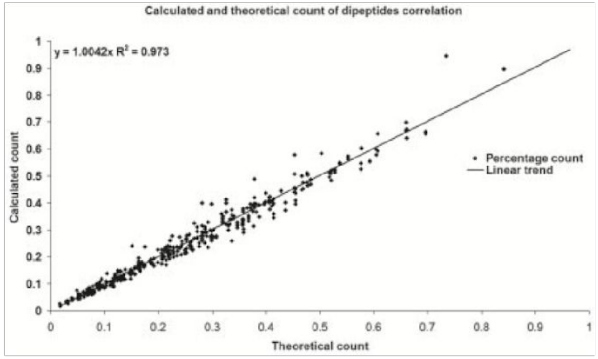


Fig. 2. The correlation between theoretical and calculated numbers (%) of dipeptides in the TREMBL database.

- key-usr—sorting by a user-defined sequence (changed in the

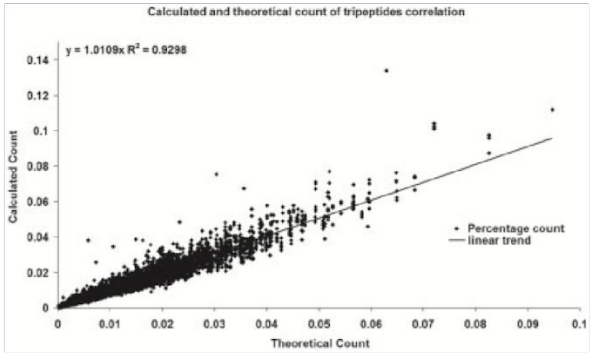


Fig. 3. The correlation between theoretical and calculated numbers (%) of tripeptides in the TREMBL database.

Table 2. Number of analyzed protein sequences within the FASTA files

Species	Proteins	Hydrolase	Polymerase	Transferase
Arabidopsis thaliana	42245	73	111	525
Danio rerio	26761	77	116	540
Escherichia coli	234128	3006	1432	10188
Homo sapiens	71093	280	267	1342
Mus musculus	48082	105	113	437
Oryza sativa	141121	307	147	1000
Saccharomyces cerevisiae	28824	103	208	436

Escherichia coli, *Homo sapiens*, *Mus musculus*, *Oryza sativa*, *Saccharomyces cerevisiae*) were searched for. The selection of species resulted from the quantity of available sequences and the

UNIX in 60 seconds...

developed in 1969 at AT&T Bell Labs

called System I, II, III, IV, V

originally an operating system, now a set of standards for operating systems (POSIX)

1972 rewritten in C

1977–1995 Berkeley Software Distribution (BSD)

by 1995, (legally) a completely rewritten version of AT&T’s UNIX and therefore able to be freely distributed

FreeBSD, NetBSD, or OpenBSD, MacOS, etc.

...UNIX in 60 seconds

1983: GNU Project launched by Richard Stallman

produced utilities and attempted to produce a kernel

1991 Linus Torvalds released the LINUX kernel

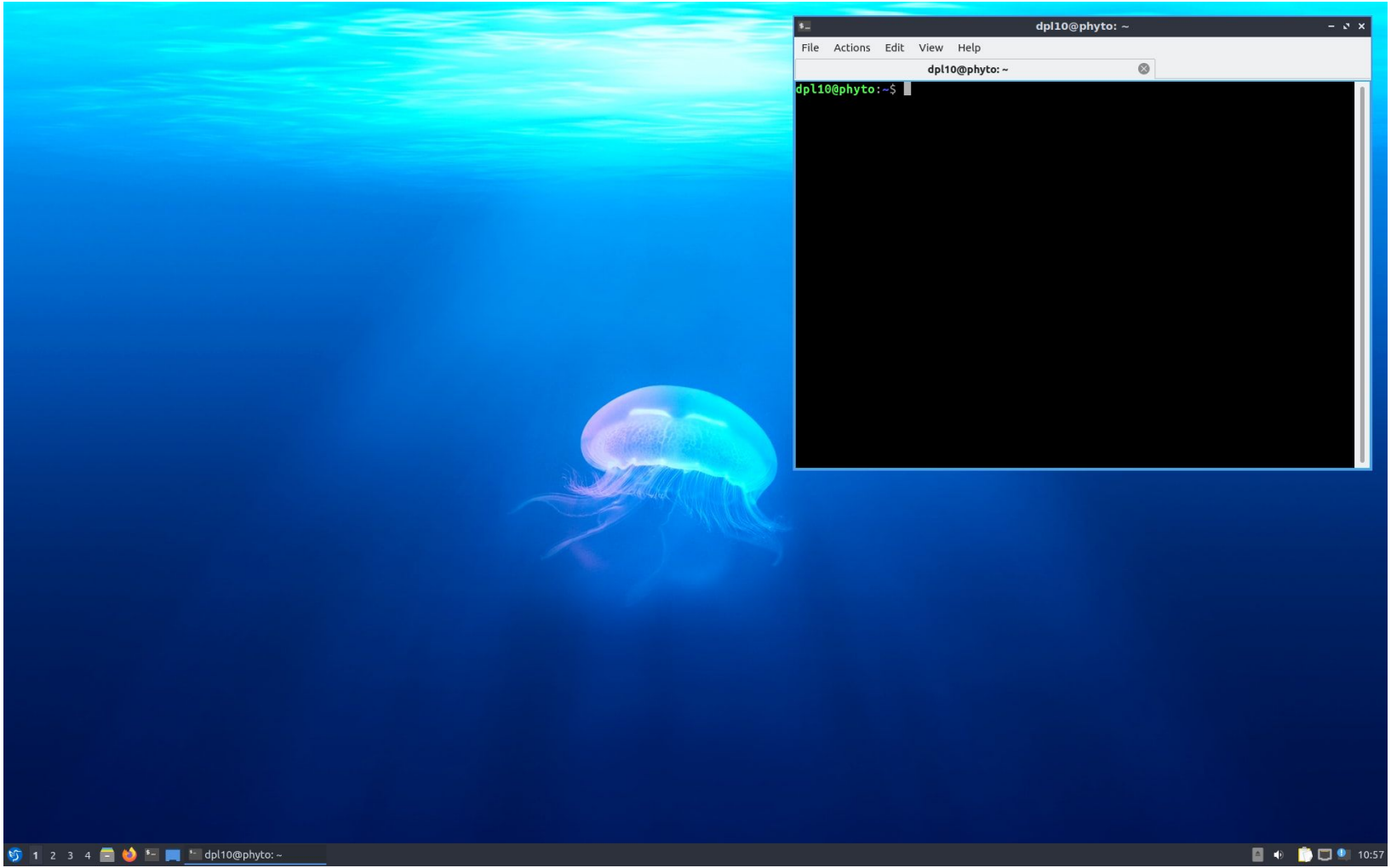
http://www.linux.org/

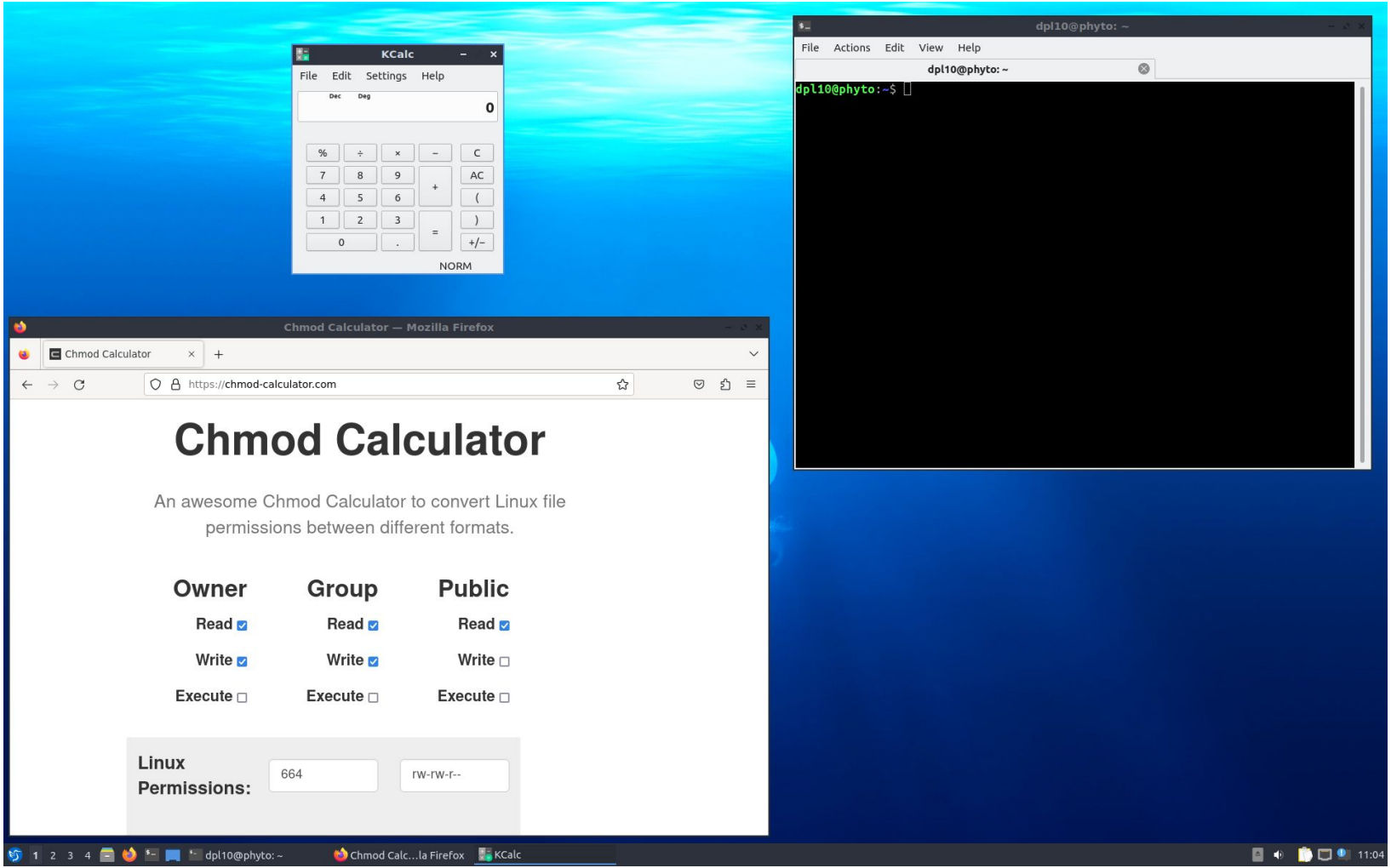
with GNU utilities => a complete operating system

many ‘LINUX’ distributions (http://distrowatch.com/)

Red Hat (Fedora, CentOS, etc.)

Debian (Ubuntu, Mint, etc.)


dpl10@phyto: ~
File
Actions
Edit
View
Help
dpl10@phyto: ~
dpl10@phyto:~$
1
2
3
4
dpl10@phyto: ~
10:57


KCalc
File Edit Settings Help
Dec Deg
0
% ÷ × − C
7 8 9 + AC
4 5 6 (
1 2 3 =)
0 . +/−
NORM
dpl10@phyto: ~
File Actions Edit View Help
dpl10@phyto: ~
dpl10@phyto:~$
Chmod Calculator — Mozilla Firefox
Chmod Calculator
https://chmod-calculator.com
Chmod Calculator
An awesome Chmod Calculator to convert Linux file permissions between different formats.
Owner Group Public
Read Read Read
Write Write Write
Execute Execute Execute
Linux Permissions:
664
rw-rw-r--
1 2 3 4
dpl10@phyto: ~
Chmod Calc...la Firefox
KCalc
11:04

‘open source’

a commitment to make software freely available

many licenses: GPL, APACHE, BSD, XII, MIT, etc.

in general:

- distribute modified versions usually allowed
- often required to distribute computer code as well
- may charge for the software
- given that the code is freely available, few purchases
- support or installation is usually purchased instead

an open source operating system

the kernel

- LINUX, BSD, freeDOS, etc.

standard utilities

- e.g. GNU

packages (via a package management system)

- e.g. apt

documentation

Ubuntu

a open source commercial product from Canonical

based on Debian LINUX

a new release every six months (April and October)

- version number is year.month
  - e.g. 16.04 was released in April 2016
- each version has a ‘cute’ code name: adjective + animal
  - e.g. Breezy Badger

the April release in even years is an LTS release

many versions based on application (e.g. server)

Lubuntu

based on Ubuntu

- uses the same package management
- just makes different utility choices

lighter weight (faster graphics, but not as pretty)

one of many ‘Ubuntu–based’ distributions

installing Lubuntu

in a virtual machine (easy to install)

- can be slow (especially for graphics)

on an external drive (easy to install)

- speed depends on the external device

on your system drive

- requires at least 32 GB free space
- can be difficult to dual boot (depending)

on an inexpensive Single–Board Computer

- can be slow (especially for disk and graphics)

how your computer works (a gross oversimplification)

hardware (e.g. processor, RAM, hard drive)

firmware (e.g. BIOS, etc.)

bootloader (e.g. GRUB)

kernel and kernel extensions (or mock kernel)

(virtual) terminal

window/display managers (e.g. XII, Aqua, Wayland), shells

programs and utilities

the user

basic UNIX concepts

UNIX assumes that you know what you are doing

will do exactly what you say (even if do not mean it)

(usually) only error messages are issued

everything is either a file or a directory

directories are really just files

data flows in streams

from user input and program output

from/to files

shells

a command–line user interface (CLI)

really just another program

- layered between the kernel and the user
- interacts with the user via stdin and stdout
- (most) execute commands in batch mode also (a script)

provides a way to interact with files (programs, etc.)

provides pipes and job control

run on the same computer as the terminal or remotely

- remote access usually via ssh

common shells…

Bourne (sh)

- released in 1977 (Stephen Bourne; UNIX version 7)
- provides minimal required POSIX features
- replaced Ken Thompson’s original UNIX shell

ash (aka dash)

- released in 1989 (Kenneth Almquist)
- an efficient open source clone of Bourne shell
- ash for BSD, dash for LINUX
- commonly used for low power computers

…common shells…

csh

- released in 1978 (Bill Joy)
- intended to be easier to use and to make sh more like C

tcsh

- released in 1983 (Ken Greer and Mike Ellis)
- csh with command–line completion and line editing

Korn (ksh)

- released in 1983 (David Korn)
- compatible with sh and includes many csh features

...common shells

zsh

- released in 1990 (Paul Falstad)
- an extended sh with many bash, ksh, and tcsh features
- now default in MacOS and Kali LINUX

bash

released in 1989 (Brian Fox)

‘**B**ourne **a**gain **sh**ell’

based on sh with features from tcsh and ksh

- most sh scripts run without modification

most commonly used shell for LINUX

variables are proceeded with $

- can be user declared or builtin (e.g. $PATH)

autoloaded scripts can be used for customization

- .bashrc