

# Praktikos darbo instrukcijos

Saulius Gražulis

Bioinformatika III

## Žymėjimai

Kodo pavyzdžiai rodomi mėlsvame fone:

```
# komanda ls -l atspausdina direktorijos (katalogo, folderio, aplanko) turinį:  
shell> ls -l  
total 8792  
-rw-r--r-- 1 saulius 109367 2008-02-10 19:43 atsarginės-dalys.odp  
-rw-r--r-- 1 saulius 16222 2008-01-13 13:26 citatos.odt  
-rw-r--r-- 1 saulius 29590 2008-02-12 22:31 darbas-01.odt  
-rw-r--r-- 1 saulius 306411 2008-02-12 22:31 darbas-01.pdf
```

Failų pavyzdžiai pavaizduoti gelsvame fone:

```
wireframe off  
select helices  
backbone 100  
color temperature
```

Dialogo su kompiuteriu pavyzdys pavaizduotas lygiapločiu šriftu (fixed-width font), tai, ką spausdinate Jūs, pavaizduota **riebiu šriftu**; kompiuteriu spausdinamas tekstas pavaizduotas paprastu šriftu. Komentarai pavaizduoti *pasvirusiu šriftu*.

## Rasmol programos instaliavimas lokaliaje direktorijoje

```
shell> cd  
  
# Komanda cd perkelia Jus į "namų direktoriją" (home directory); mes ją čia  
# įvykdome tam, kad garantuotai atsidurtume "namuose", nepriklausomai nuo  
# vykdytų komandų istorijos.  
  
shell> mkdir bin; cd bin  
shell> wget http://www.bernstein-plus-sons.com/cgi-bin/yaya/decomp.cgi/software/RasMol\_2.7.2.1.1/RasMol.LINUX/RedHat\_7/i386/rasmol\_32BIT  
  
# Galime naudoti bet kokią grafinę (GUI) naršyklę, bet wget komanda atliks  
# tą patį darbą ir ją galime nesunkiai užrašyti kaip pavydį šioje  
# instrukcijoje ;)  
  
shell> chmod 755 rasmol_32BIT  
shell> ln -s rasmol_32BIT rasmol  
  
# Komanda ln sukuria nuorodas – "kietas" (hard links) arba simbolines  
# (symbolic links, soft links). Nuorodos – tai failai, kurie rodo į kitų failų  
# turinį. Mums geriau tinka simbolinė nuoroda, nes ateity visada matysime,  
# kurlink ji rodo, tad rekomenduojų 'ln -s' komandą, nors "kieta" nuoroda,  
# sukurta komanda ln veiks panašiai.  
  
shell> ls -l  
total 448  
lrwxrwxrwx 1 saulius 12 2008-02-19 20:20 rasmol -> rasmol_32BIT  
-rw-r--r-- 1 saulius 452252 2008-02-19 20:20 rasmol_32BIT  
  
shell> cd; mkdir doc; cd doc  
  
shell> wget http://www.bernstein-plus-sons.com/cgi-bin/yaya/decomp.cgi/software/RasMol\_2.7.2.1.1/doc/rasmol.hlp
```

```

shell> cd
tcsh> echo 'setenv PATH ${HOME}/bin:${PATH}' >> ~/.cshrc
# arba
tcsh> echo 'path = ( $path ${HOME}/bin )' >> ~/.cshrc
# arba, jei naudojate bash apvalkalėli:
bash> echo 'export PATH=${HOME}/bin:${PATH}' >> ~/.bashrc
shell> which rasmol
shell> rasmol

```

## Molekulių analizė Rasmol programa

```

shell> cd bin; wget http://vartai.ibt.lt/~saulius/scripts/wgetentry
shell> chmod 755 wgetentry; cd
shell> mkdir darbas-01; cd darbas-01; ln -s ~/doc/rasmol.hlp .
shell> wgetentry lknv
shell> rasmol lknv.pdb
RasMol> help
RasMol> wireframe off
RasMol> select helices
Rasmol> save "lknv-helices.pdb"
# Kitas koordinacijų užrašymo į failą variantas:
RasMol> write pdb "lknv-helices.pdb"
RasMol> backbone 100
RasMol> color temperature
RasMol> write "lknv-helices.pnm"
RasMol> save ppm "lknv-helices.ppm"
# arba
RasMol> write "lknv-helices.ppm"
RasMol> quit
shell> pnmtopng lknv-helices.ppm > lknv-helices.png; rm -v lknv-helices.pnm

```

## Rasmol skriptai

Paleiskite Jūsų mėgiamą redaktorių; pavyzdžiui, **kate**:

```
shell> kate helices.rasm &
```

ir sukurkite tokio turinio failą:

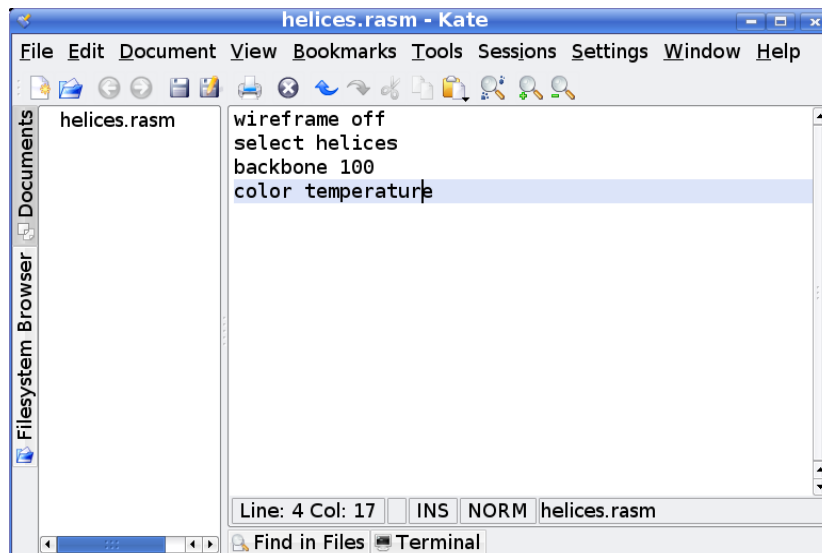
```

wireframe off
select helices
backbone 100
color temperature

```

Dabar galite paleisti Rasmol, nurodę jam vykdyti šį skriptą, užuot spausdinę komandas kiekvieną kartą Rasmol lange:

```
shell> rasmol -script helices.rasm lknv.pdb
```



## Shell-skriptai su Rasmol'u

Ką tik parašytas skriptas turi vieną trūkumą – jis vienodas visiems analizuojamiems PDB failams. Jame nenumatyta galimybė nei perduoti parametrus (nurodančius, pvz., kokius molekulos elementus pasirinkti ar kokią vaizdavimo schemą naudoti), taip pat nenumatyta galimybė nurodyti išėjimo (išvesties) failo vardą, jei nutartume paveikslėlių arba pasirinktus atomus užrašyti į diską.

Minėtas funkcijas galima nesunkiai įgyvendinti Unix ar Linux shell programavimo kalba. Sukurkite, naudodami tekstų redaktorių, tokį failą:

```
#!/bin/sh

INPUT_PDB="$1"
OUTPUT_PDB="$2"

rasmol -nodisplay ${INPUT_PDB} <<EOF
#
# Rasmol "here document"
#
select helices
write pdb "${OUTPUT_PDB}"
exit
EOF
```

Išsaugokite failą vardu „pdb-sel-helix.sh“. Dabar galite paleisti šį skriptą, nurodydami bet kokius įėjimo ir išėjimo failus – shell'as juos „įdės“ į reikiamą vietą:

```
shell> chmod 755 pdb-sel-helix.sh
shell> ./pdb-sel-helix.sh 1knv.pdb 1knv-helices.pdb
```

Turėdami savo procedūrą (algoritma) užrašytą faile, galite jį automatiškai pritaikyti dideliame turimų failų kiekiui:

```
shell> mkdir inputs; cd inputs; wgetentry 1knv 2c1l 1fiu; cd ..
shell> mkdir outputs
bash> for i in inputs/*.pdb; do BASENAME=`basename $i`; ./pdb-sel-helix.sh $i
outputs/$BASENAME.pdb; done
```

## Perl'o skriptai su Rasmol'u

Jei Jūs geriau žinote Perl'ą, negu shell'ą, galite padaryti Rasmol automatizavimo skriptą ir Perlu:

```
#!/usr/bin/perl -w

use strict;

my $input_pdb = $ARGV[0];
my $output_pdb = $ARGV[1];

open( RASMOL, "|rasmol -nodisplay ${input_pdb}" )
  or die( "$0: could not call Rasmol: $" );

print RASMOL <<EOF;
select helices
write pdb "${output_pdb}"
exit
EOF

close( RASMOL ) or die( "$0: error running Rasmol: $" );
```

## Makefile'ai

Dažniausiai tenka failus su skriptais arba parametrais keletą (daug) kartu redaguoti ir kiekvieną sykį suskaičiuoti rezultatus su naujais skriptais. Tokiu atveju naudinga turėti Makefile'ą ir naudoti programą make rezultatams perskaičiuoti. Programa make pasakys, ar reikia ką nors skaičiuoti, ar ne, automatiškai atliks tik tuos skaičiavimus, kuriuos būtina atlikti. To, kas jau suskaičiuota, make sistema papildomai neskaičiuos.

Makefile'as gali atrodyti taip:

```
INPUT_DIR = inputs
OUTPUT_DIR = outputs

INPUT_PDB_FILES = ${wildcard ${INPUT_DIR}/*.pdb}
OUTPUT_PDB_FILES = ${INPUT_PDB_FILES:${INPUT_DIR}/%.pdb=${OUTPUT_DIR}/%-
helices.pdb}

LOG_FILES = ${INPUT_PDB_FILES:${INPUT_DIR}/%.pdb=%.log}

.PHONY: all clean cleanAll

all: ${OUTPUT_PDB_FILES} ${LOG_FILES}

${OUTPUT_DIR}/%-helices.pdb: ${INPUT_DIR}/%.pdb
    ./pdb-sel-helix.sh $< $@ | tee $*.log

%.log: ${OUTPUT_DIR}/%-helices.pdb;

clean cleanAll:
    rm -f ${OUTPUT_PDB_FILES}
    rm -f ${LOG_FILES}
```

Programa make naudojama taip:

```
shell> cd darbas-01
shell> make
shell> make clean
shell> make all
```

## **Subversija**

Visus pakeitimus naudinga registruoti ir išsaugoti. Tam gerai tinka versijų kontrolės sistemos. Subversion – viena lanksčiausių ir patogiausių versijų kontrolės sistemų:

```
shell> cd darbas-01
shell> svn co svn://saulius-grazulis.lt/praktikos-darbai/Saulius_Grazulis .
shell> svn update
shell> svn status
shell> svn-add-unknown
shell> svn ci
shell> svn up
shell> svn st
```

Nepamirškite taško („.“) komandoje „co“, jei norite paversti savo jau turimą darbinę direktoriją su pradėtais projekto failais į Subversijos darbinę kopiją!