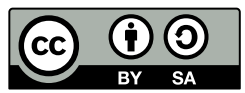
Programų testavimas

Saulius Gražulis

2010 ruduo

Vilnius University, Faculty of Mathematic and Informatics
Institute of Informatics



Kam rekalingas testavimas?

Beware of bugs in the above code; I have only proved it correct, not tried it.

Donald Knuth
http://www-cs-faculty.stanford.edu/ knuth/faq.html
2009.11.09

Parašykime paprastą programėlę

- Užduotis: duota DNR seka faile; sugeneruokim komplementarią grandinę
  - rekia „apversti“ grandinę (nes visada vaizduojame DNR $5' \to 3'$ kryptimi)
  - rekia pakeisti bazes į komplementarias
- Simbolių eilutę galime apversti Perlu:

```
perl -lne 'print reverse(split(""))'
```

- Pakeisti bazių raides į komplementarių bazių raides galime komanda `tr`:

```
tr "ATCG" "TAGC"
```

„Supakuokime“ komandas į SH skriptus

tests/revert-2009.11.09/bin/revertlines:

```
#! /bin/sh

# Print each line in the input files reversed
# (i.e. from last char to the first one).

set -ue

perl -lne 'print reverse(split(""))' "$@"
```

„Supakuokime“ komandas į SH skriptus (2)

`tests/revert-2009.11.09/bin/complement-dna-bases`:

```
#! /bin/sh

# Change a letter for each DNA base to the
# complement one.

set -ue

tr "ATCG" "TAGC"
```

Paleiskim programą bent vieną kartą

- Galime paleisti abi programas tiesiog komandinėje eilutėje:

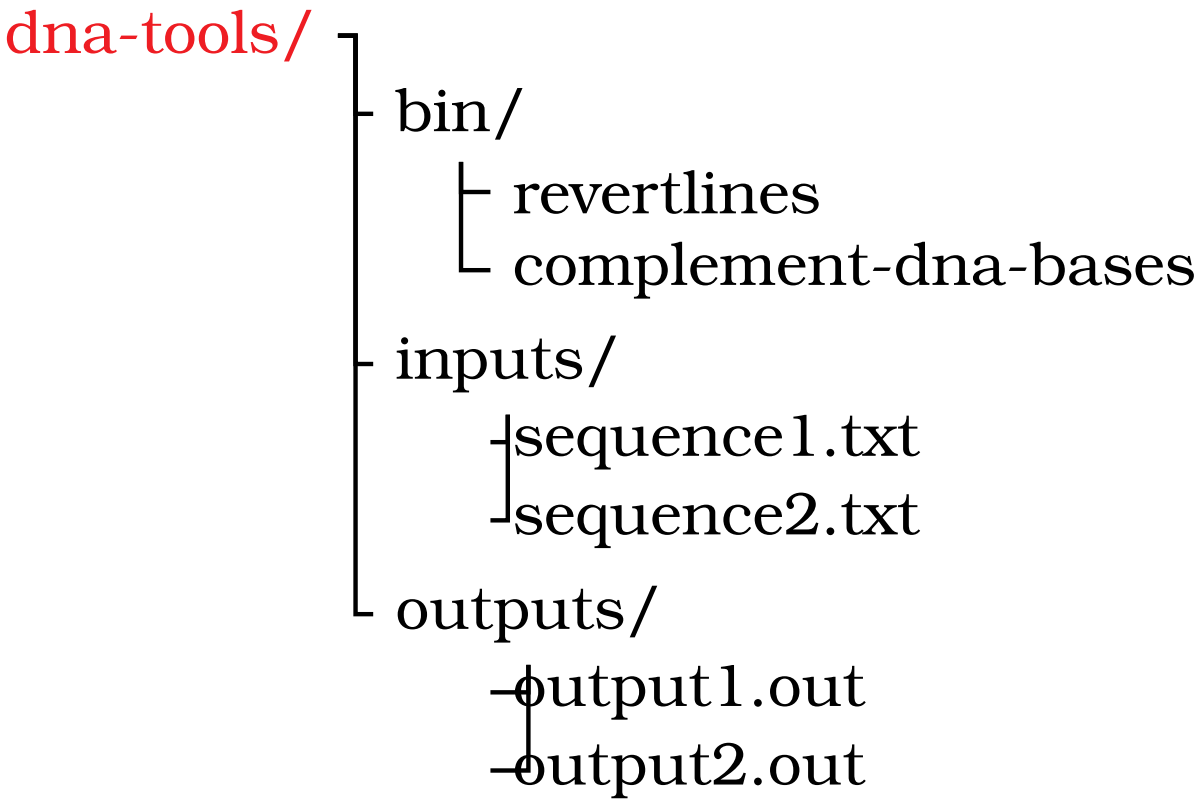
```
sh> echo CAATTG | ./bin/revertlines
GTTAAC

sh> echo CAATTG | ./bin/revertlines \
| ./bin/complement-dna-bases
CAATTG
```

- Bet lieka dar klausimų:
  - ar teisingai algoritmas veiks, jei duomenys ateis iš failo, o ne iš STDIN?
  - ar teisingai algoritmas veiks nepalindrominėms sekoms?
  - kaip programos reaguos į neteisingus duomenis (pvz. neegzistuojančius failus?)
  - kai pakeisim kurią nors programą, ar senos funkcijos liks nesugadintos?

Testų automatizavimas

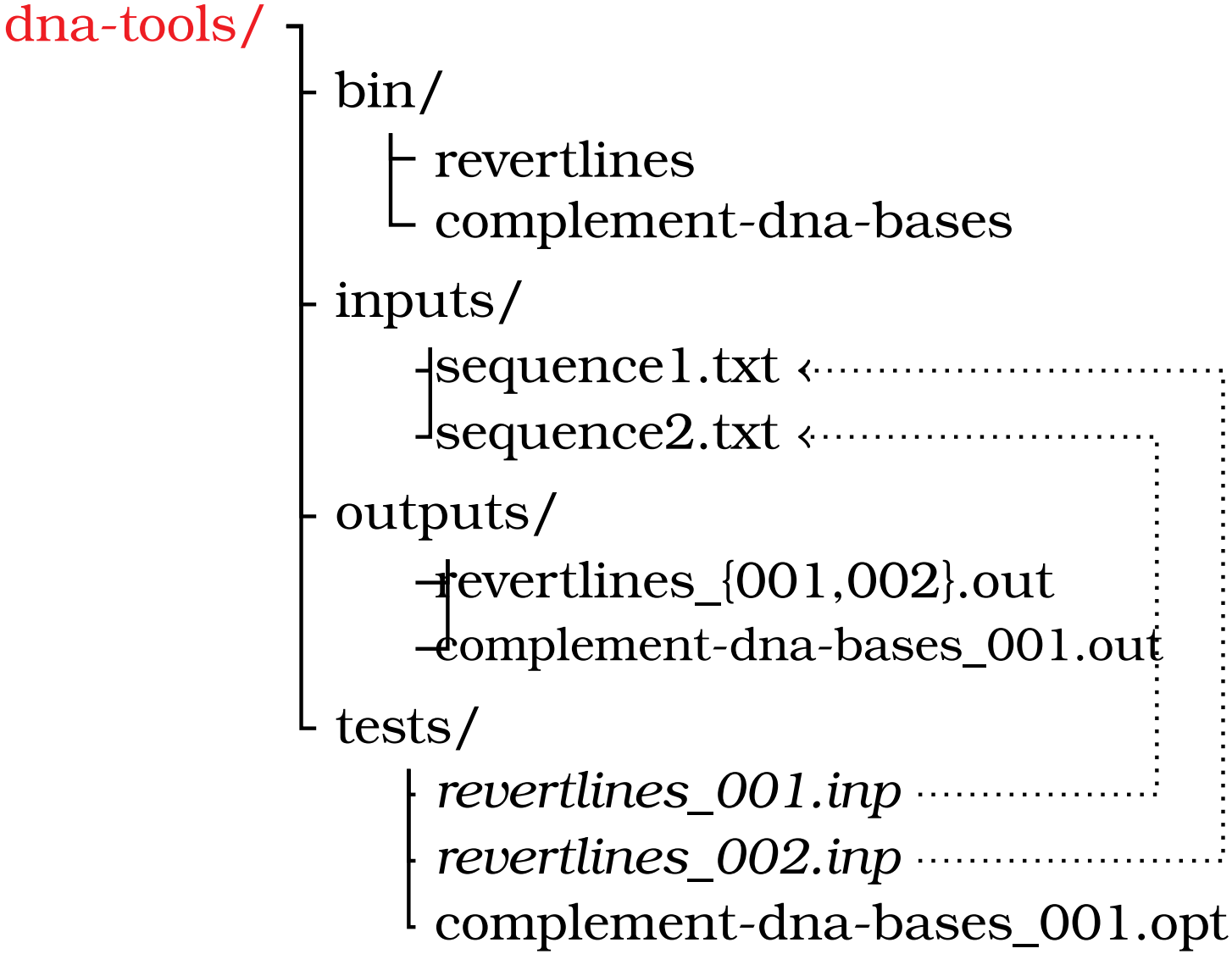
Užsirašykime testų duomenis ir laukiamus rezultatus į failus



- Problemėlės:
  - kaip surasti, kuris testas kuriam skriptui skirtas
  - kaip paleisti visus testus automatiškai.

Testų automatizavimas (2)

Paruoškime kiekvienam testui atskirą failą – nuorodą



Simbolinės nuorodos (symbolic links)

Informacja ten kur jos rekia, neeikvojant vietos...

- Simbolinė nuoroda sukuriama komanda:

```
sh> ln -s failas.txt nuoroda.txt
```

- Simbolinė nuoroda atsimena, kurlink ji rodo:

```
sh> ls -l nuoroda.txt
lrwxrwxrwx 1 saulius 10 2009-11-10 12:45 nuoroda.txt -> failas.txt
```

- Visom „paprastoms“ programoms nuoroda atrodo kaip failas, turintis tą pačią informaciją, kaip ir originalas:

```
sh> head failas.txt nuoroda.txt
==> failas.txt <==
tekstas

==> nuoroda.txt <==
tekstas
```

Makefile'as automatiniam testavimui

tests/2revert-2009.11.09/Makefile:

```
# Makefile for running automated tests

BIN_DIR = bin

TEST_DIR = tests
TEST_EXT = inp
OPT_EXT  = opt

OUTP_DIR = outputs
OUTP_EXT = out
DIFF_EXT = diff

TEST_CASES = $(wildcard ${TEST_DIR}/*.${TEST_EXT})

TEST_OUTPUTS = ${TEST_CASES:${TEST_DIR}/%.${TEST_EXT}=\
        ${OUTP_DIR}/%.${OUTP_EXT}}

TEST_DIFFS = ${TEST_CASES:${TEST_DIR}/%.${TEST_EXT}=\
        ${OUTP_DIR}/%.${DIFF_EXT}}
```

Makefile'as automatiniam testavimui (tęsinys)

tests/2revert-2009.11.09/Makefile:

```
TESTER = tools/run-test

.PHONY: all clean distclean cleanAll test tests out outputs

all: test

test tests: ${TEST_DIFFS}

outputs out: ${TEST_OUTPUTS}

${OUTP_DIR}/%.${DIFF_EXT}: ${TEST_DIR}/%.${TEST_EXT}
        @${TESTER} ${BIN_DIR}/$(shell echo $* | sed -e 's/_[0-9]*$$//') \
                $@ \
                ${@:%${DIFF_EXT}=%${OUTP_EXT}} \
                $<

${OUTP_DIR}/%.${OUTP_EXT}: ${TEST_DIR}/%.${TEST_EXT}
        if [ ! -f $@ ]; \
        then \
                ${BIN_DIR}/$(shell echo $* | sed -e 's/_[0-9]*$$//') $< > $@ 2>&1; \
        fi

clean:
        rm -f ${TEST_DIFFS}

cleanAll distclean: clean
```

Testų paleidimo skriptas

tests/2revert-2009.11.09/tools/run-test:

```
#! /bin/sh

# Run a test, report its result

# Usage:
# run-test script test_001.diff test_001.out test_001.inp

set -u

SCRIPT="$1"
DIF_FILE="$2"
OUT_FILE="$3"
INP_FILE="$4"

printf "%-30s: " $(basename ${INP_FILE})

${SCRIPT} ${INP_FILE} \
| diff ${OUT_FILE} - \
> ${DIF_FILE} 2>&1

if [ $? = 0 ]
then
    echo OK
else
    echo FAILED:
    cat ${DIF_FILE}
fi

true
```

Kitos testavimo sistemos

Yra iš ko pasirinkti... :)

- Perl
  - Test::Unit (a.k.a. PerlUnit)
  - Test::More
- Java/Web
  - Selenium
- Python
  - PyUnit
  - py.test

http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks
2010-11-29