Phobos

Version 3.3.2

A tandem repeat search program

Christoph Mayer

2007

Disclaimer:
The Phobos-program (version 3.3.2) is distributed "as is" and in the hope that it will be useful, but it comes without warranty of any kind. The program is still under development. Please report any crashes, bugs, or problems you discover. This manual is still under construction. Therefore, some features of Phobos are only briefly documented in this manual.

New versions of Phobos and of this manual will be made available from the following web-page: `http://www.ruhr-uni-bochum.de/spezzoo/cm/cm_phobos.htm`.

Contents

Chapter 1

Introduction

Phobos is a powerful program to search for tandem repeats (also referred to as *vntr* or *satellites*) in DNA sequences and complete genomes.

Features of Phobos:

- Phobos can detect tandem repeats in three different search modes: (i) Exact tandem repeats. (ii) Imperfect tandem repeats for which at least two successive units are identical. (iii) General imperfect satellites for which no two units need to be identical.
- It searches for all tandem repeats that have a repeat unit in a specified unit size range. In particular, Phobos does not require a pattern library.
- It can search for *exact tandem repeats* with a unit size of 1 to several thousand base pairs. The upper unit size is only limited by the computer resources. On complete genomes, such as the human genome, a unit size of 10000 bp is accessible in a reasonable amount of time. Furthermore, Phobos is able to detect overlapping satellites and/or sub-satellites where a satellites with a shorter repeat unit is nested in a larger satellite with a larger unit.
- It can detect *imperfect tandem repeats* with a unit size of 1 to several dozen base pairs. Imperfect tandem repeats (sometimes also referred to as approximate tandem repeats) can contain mismatches and gaps (indels). No two units must be equal to be detected, which is of particular interest for larger repeat units. Phobos is able to detect overlapping imperfect satellites, sub-satellites and also to find alignments with alternative repeat units.
- It uses an exact and in particular non-probabilistic search algorithm, which has the advantage of a higher accuracy of search results, especially for repeats with few repeat units in length. The exact search algorithm makes Phobos the best available tool for tandem repeat statistics in genomes.
- It uses the alignment score as an optimality criterion for the question of how far a tandem repeat should be extended in both directions. Mismatch and gap penalty can be specified independently, which provides a clear and easily tractable scoring scheme.
- The minimum score and length requirements for of a tandem repeat can be specified in a very flexible way.

- The output format of Phobos can be adjusted very flexible. If requested, Phobos can report flanking regions and provide alignments of the subject sequences against the putative perfect repeat units. Repeat units can be reported "as they are" or in a normalized form.

Command-line (CL) versus graphical user interface (GUI):
Phobos is available in form of a *CL program* as well as a *GUI-program*. The graphical user interface, however, comes at the price of a slightly reduced performance. The main advantage of the CL-program is that analyses can be started automatically from batch files or from other programs.

Platforms:
The CL- and GUI-version of Phobos are available for Mac OS X, Linux, and Windows.

System requirements:
For the analysis of a complete genome of the size of the human genome, Phobos requires 2 to 4 GByte of RAM, depending on the search parameters. If Phobos has to search for particularly short repeats, the memory requirements increase. A processor with at least 2 GHz is recommended.

About Phobos:
Phobos has been developed and implemented by Christoph Mayer. It is completely written in C++. On all platforms it has been compiled, with the GNU gcc compiler. The *CL program* uses the "Templatized C++ Command Line Parser Library" which is distributed under the MIT License. The *GUI-program* uses FLTK, version 1.1.7, distributed under the Library General Public License.

1.1 Obtaining and Installing the Program

1.1.1 Optaining Phobos

Phobos can be downloaded from the authors web page:
`www.rub.de/spezzoo/cm/cm_phobos.htm`.

1.1.2 Installation

1. Unpack the zip-archiv you have downloaded.
2. Move the Phobos binary executables, located in the *bin* directory, to a destination of your choice.

   A good place to put the GUI-program is a *Program* or *Application* folder. The GUI-program is started by clicking on its item.

   A good place for the CL-program is somewhere in your systems path, e.g. the */usr/local/bin* directory on Unix machines. This allows all users to start the program form the command-line from every directory they are in. The most practical way to start Phobos in this case would be to navigate, on the command-line, to your data directory and start Phobos directly from there.

Chapter 2

Using Phobos

2.1 Using Phobos with GUI

The GUI-version of Phobos provides the same functionality as the CL-version. Unfortunately, the additional comfort comes at the price of a noticable performance reduction. Therefore, if computational time is of concern, the CL-version of Phobos might be preferred.

The GUI-program can be started by clicking on its item. Starting it from a terminal window is also possible, but command-line parameters passed to it are ignored.

2.1.1 Phobos remembers search parameters

The first time Phobos is started on a computer, default search and output parameters are being displayed in the main Phobos window. At the end of each Phobos session, all parameters as well as the names of the input and output files are being saved automatically. They will be used as startup values the next time Phobos is being used.
A set of parameters together with the names of the input and output files can also be saved and loaded from a user specified preference file. This is described in Section 2.1.3 below.

2.1.2 Choosing an input and output file

Phobos can read molecular sequence files in the FASTA format. Input and output files can be entered directly in the text fields or, more convenient, can be selected in a file chooser which is invoked by clicking on the corresponding *Browse* button. Every time an input file has been selected with the file chooser, Phobos suggests a name for the output file which has the same location and name as the input file, but with the *.phobos* extension substituted for the original extension.

2.1.3 The main Phobos control buttons

The main Phobos control buttons are situated in the upper right hand corner. They are:

Load parameters: Load a set of search settings that have previously been saved.

Save parameters: Save the current search settings in a file for future reference.

Reset parameters: Reset the search and output parameters to default values.

Run analysis: Start the analysis with the current search settings.

About: Show Phobos version number, copyright, author, and acknowledgments.

Quit: Quit Phobos.

Several of these actions require that the current search settings are valid, see Section 2.1.5.

2.1.4 The file chooser in Phobos

The file chooser in Phobos provides two particularly interesting features:

- a file preview window,
- the ability to add and manage data directories that are frequently used in a *Favorites* menu.

2.1.5 Validity of search settings is checked automatically

Every time an attempt is made to *run an analysis*, *save the parameter values*, or to *quit Phobos* the validity of the search settings is checked. In case these parameters are not valid, Phobos sets them to default values and aborts the last command. In particular, and with one exception, parameter input fields are not allowed to be blank, but must contain a valid value. Only the *from sequence* and/or *to sequence* input fields can be left bank, which the program translates, respectively, to the number of the first and last sequence.

2.1.6 Search and output parameters

The search and output parameters of Phobos are described in detail in Section 3.4 and 3.5.

2.2 Using Phobos on CL

Using Phobos from the command-line has two advantages: (i) the program runs faster and (ii) it can be run from within scripts, e.g. on computer clusters, or it can be called from other programs.

On the command-line, Phobos runs non-interactive, i.e. all search parameters have to be passed to it as command-line arguments.

2.2.1 Required and optional command-line arguments

Specifying the input and output file:
Phobos has one required argument, which is the name of the input file in the FASTA format. All other arguments are optional and alter the default search and output behavior. If a second file name is passed to Phobos, it is used as the output file. Otherwise, results are written to the standard output.

Examples:

```
phobos_cl input-file
phobos_cl input-file output-file
```

The first call to Phobos starts a default analysis of the sequences in the file called "input-file" and writes the results to the standard output. The second call to Phobos writes the results to the file called "output-file".

Search and output options:
Program options are command-line arguments which start with a "-" or "--" followed by the option name. Some options work as switches whereas others require an option value. Note that some options have two alternative option names, a *long* and a *short* version. A complete list of command-line options is given in the following.

General options:

-v, --version
Displays version information and exits.

-h, --help
Displays usage information and exits.

Search options:

--searchMode exact, -M exact
Search for exact repeats.

--searchMode extendExact, -M exact
Perfect repeat seeds are extended to imperfect repeats.

--searchMode imperfect, -M imperfect
Search for imperfect repeats. *The default.*

--minUnitLen <int>, -u <int>
Minimum unit (pattern) size in search. Default: 1

--maxUnitLen <int>, -U <int>
Maximum unit (pattern) size in search. Default: 6

--firstSeq <int>
Number of first sequence to be processed in this analysis.

--lastSeq <int>
Number of last sequence to be processed in this analysis.

--indelScore <integer>, -g <integer>
Gap score. Typical: -4 to -6. Default: -6

--mismatchScore <int>, -m <int>
Mismatch score. Typical: -4 to -6. Default: -6

--recursion <int>, -r <int>
Recursion depth. Typical 3 to 7. Default: 5

--maximum_score_reduction <int>, -R <int>
Maximum score reduction allowed in search. Typical 30. Default: switched off

--minScore <int>, -s <int>
See section 3.4. Default: 8

--minScore_a <int>
See section 3.4. Default: 0

--minScore_b <float>
See section 3.4. Default: 0

--minLength <int>, -l <int>
See section 3.4. Default: 0

--minLength_a <int>
See section 3.4. Default: 0

--minLength_b <float>
See section 3.4. Default: 0

--minPerfection <float>, -P <float>
Minimum perfection of a repeat. Default: 0. *Not implemented in the current version.*

--NsAsMissense
Treat N's as missense. Default: Treat N's as neutral with score 0.

-N <int>, --succN <int>
The maximum number of successive N's allowed in a satellite. Default: 2.

Output options:

-D, --dontRemoveMostlyOverlapping
Don't remove repeats that are mostly overlapped by or do mostly overlap a higher scoring repeat.

-f <int>, --flanking <int>
Number of flanking nucleotides to be shown.

--maskX
Write masked sequence into file with ".masked" extension.

--outputFormat <int>
0: Phobos format, 1: extended Phobos format, 2: sputnik compatible format.

--printRepeatSeqMode <int>
0: don't print sequence, 1: print sequence, 2: print alignment.

--NPerfectionMode <int>
0: asMismatch, 1: asNeutral, 2: asMatch. Default: 0

--reportUnit <int>

0: asIs. 1: Alphabetical normal form given by the alphabetically minimal string among all cyclic permutations of the unit. 2: Alphabetical normal form given by the alphabetically minimal string among all cyclic permutations of the repeat unit or its reverse complement. Default: 2

Chapter 3

How Phobos searches, scores and reports tandem repeats

3.1 Input file format

Phobos can read molecular sequence data in the FASTA format. If a file contains more than one sequence, tandem repeats are searched separately in each sequence. Phobos does not concatenates the sequences. The length of sequence names is practically unlimited.

3.2 How Phobos searches for tandem repeats

At each position in a sequence and for each unit length, Phobos checks whether this position is a valid starting point. If this is the case, Phobos tries to extend the repeat in both directions as far as possible. By this strategy, no repeat library is used. Phobos also searches at positions where another repeat has already been found. This allows Phobos to find hidden or overlapping satellites, as well as to find alignments with different repeat units in order to find the alignment with the highest score. If the scoring scheme for instance allows to interpret an $(\mathrm{ATATAG})_{10}$ repeat as an imperfect dinucleotide repeat, Phobos also finds the better alignment with the hexanucleotide repeat unit. Of course, Phobos also takes care that any repeat, e.g. $(\mathrm{AT})_{40}$, is not reported as a repeat with unit length being a multiple of the true unit length. (The search algorithm will be described in more detail in an upcoming publication.)

3.3 Scoring scheme and optimality criterion

The score of a tandem repeat with a given start and end point in the subject sequence is the score of the best local alignment that has been found with successive copies of a putative repeat unit. Phobos computes the score as follows: Each match in the alignment gets a positive score of 1. Mismatch and gap (indel) score can chosen by the user and are only restricted to be a negative integers. The starting unit is not scored. (In one of the next versions of Phobos this will be changed such that the user can choose whether the starting unit is scored or not.) Furthermore, the user can decide whether N's are treated as neutral with score 0 or as missense, i.e. as mismatch or gap.

Phobos uses the score of a tandem repeat as its optimality criterion, i.e. a repeat is treated as better than another repeat if its score his higher. This optimality criterion is primarily used to decide whether a repeat should be extended beyond a mismatch or gap position or not. Phobos extends a repeat, if a longer repeat can be found that has a higher score. In case a longer repeat has the same score as a shorter one, the shorter repeat is preferred.
The score of a repeat is also important if the "remove mostly overlapping" option is in effect. In this case, Phobos checks whether any repeat is covered by another repeat or whether their overlap is almost as long as one of the repeats themselves. If such an overlap is found, Phobos removes the repeat with the smaller score independent of which repeat is longer.

3.4 Detailed description of search parameters

3.4.1 Search modes

Phobos provides three major search modes: (i) imperfect search, (ii) prefect search, and (iii) extend exact search.

Imperfect search
Phobos tries to find tandem repeats by allowing mismatches and gaps (see the previous section for details). In this mode no two units need to be identical to find the repeat.

Perfect search
Phobos searches for exact repeats. If N's are treated as neutral (i.e. not as missense, see below), they are allowed in this mode.

Extend exact search
Phobos first searches for perfect tandem repeat *seeds* and extends these to both sides to imperfect repeats by allowing mismatches and gaps. The minimal seed size is 5 bp for mono-, 6 bp for di-, and 7 bp for trinucleotide repeats. For repeats with longer units at least 2 exact units must be present.

3.4.2 Save masked sequences

With this option in effect, Phobos does not only report the tandem repeats it found, but also saves the molecular sequences in the input file in a new file with all tandem repeats masked by X's. The new file is saved in the same directory as the input file. It has the same name as the input file with ".masked" appended to it.

3.4.3 Analyze sequence range

If not all sequences in a file need to be analyzed, the range of sequences that should analyzed can be specified.

3.4.4 Treat N's as missense

The default is to treat N's as neutral characters with a score of zero. With this option in effect, N's are treated as missense positions leaving the decision to Phobos whether it is a mismatch or gap position.

3.4.5 Maximum number of successive N's

Particularly if N's are treated as neutral characters with a score of zero, the maximum number of successive N's must be limited. The specified maximum number of N's has an effect regardless of whether N's are treated as neutral or as missense. No N's are allowed at the beginning or the end of a tandem repeat.

3.4.6 Mismatch and gap score

In the alignment of a putative tandem repeat with its exact counterpart each match gets a score of 1. The mismatch and indel score can be specified independently by the user. See Section 3.3 for details on the scoring scheme.

3.4.7 Recursion depth

Phobos uses a recursive alignment algorithm which will be explained in more detail in an upcoming publication. A higher recursion depth leads to a higher quality of the alignment. An alignment of high quality is obtained with the default recursion depth of 5. A recursion depth of 3 leads to a low alignment quality, whereas a very high alignment quality is obtained with a value of 7. It should be noted, that values of the recursion depth of Phobos can not be compared to that of sputnik or SciRoko. For the same value of the recursion depth, Phobos produces a significantly higher alignment quality than these two programs.

3.4.8 Maximum score reduction

If the maximum score reduction is switched off, a search (in the current search direction) for a possibly longer and better alignment of a putative repeat is not stopped before the score falls below the value $(-1) \times$ (unit size). If a maximum score reduction is specified, the search will be stopped if the score falls below the maximum value found during the search lowered by the value of the *maximum score reduction*. With this option, Phobos runs significantly faster, with the drawback of a reduced sensitivity in the tandem repeat search.
A value of 30 for the maximum score reduction, together with not too high a recursion depth, allows Phobos to search for imperfect tandem repeats with a unit size range of 1 to several thousand base pairs in complete genomes.

3.4.9 Minimum score and length

Tandem repeats must fulfill minimum requirements concerning their score *and* length to be reported by Phobos. Both threshold values, the minimum score and the minimum length are each controlled by three parameters to yield a minimum value that depends on the unit length of the tandem repeat. Using the parameter names of the command line program, these parameters are referred to as `minScore`, `minScore_a`, and `minScore_b` as well as `minLength`, `minLength_a`, and `minLength_b`. Given these values, the unit length dependent minimum score $S(ul)$ and minimum length $L(ul)$ are computed according to the following formulas, where ul is the current unit length

$$S(ul) = \max(\,\text{minScore},\ \text{minScore\_a} + \text{minScore\_b} * ul) \tag{3.1}$$

$$L(ul) = \max(\,\text{minLength},\ \text{minLength\_a} + \text{minLength\_b} * ul) \tag{3.2}$$

Thus, the unit length dependent minimum score is given either by the constant value `minScore` or by the value of the term (minScore_a + minScore_b$*ul$) depending linearly on the unit length, whichever of the two is higher. An analogous interpretation holds for the minimum repeat length. It should be noted that the `minScore`, `minScore_a`, `minLength`, and `minLength_a` parameters must be integer values, whereas the `minScore_b` and `minLength_b` can be floating point numbers.
A tandem repeat is reported by Phobos if its score and its normalized length (given by equation 3.4 below) are both equal or higher to the corresponding threshold values.

3.5 Detailed description of output parameters and format

In each analysis, where Phobos analyses the sequences of a single input file, results are written to a single output file or the console. Several program options influence the output format.

Each Phobos output starts with an "analysis header" which summarizes some general informations such as the Phobos version that created the output and a complete list of all search settings. In this header section, each line starts with the "#" symbol, so that they it can be recognized as comment lines by human readers and parser programs.
Every time Phobos starts to analyze a new sequence of the FASTA file, it writes two lines of output: The sequence name in FASTA format (proceeded by a ">" symbol) in the first line and the sequence length in the second line.

Example:

```
>
sequence length: 131
```

The way tandem repeats are reported to the output file depends on several output options.

3.5.1 General output format

There are two output formats for the general repeat information: The *Standard Phobos format* and the *Extended Phobos format*.

Standard Phobos format:

Example:

```
tetranucleotide      296 :      331 |    36 bp |    37 BP |    19 pt |  94.595 % | unit AAAG
```

The columns describe, respectively, the repeat class, the start and end positions in the subject sequence, the length of the tandem repeat in the subject sequence, the normalized repeat length (see equation 3.4 below), the score, the percentage perfection and finally the repeat unit.

Extended Phobos format:

Example:

```
tetranucleotide      296 :      331 |    36 bp |    37 BP |    19 pt |  94.595 % |   1 mis |   0 ins |   1 del |   0 N |  unit AAAG
Missense: (9,M),(22,D),
```

In this format we have additional columns for the number of mismatch, insertions, deletions and N (if they are not treated as missense) positions. Furthermore, a second line lists the missense positions together with the type of the missense.

Normal and normalized repeat length:
Phobos prints two repeat lengths: The length of the tandem repeat in the subject sequence and a normalized repeat length. If *start* and *end* are the first and last position of the tandem repeat in the subject sequence the repeat length is

$$len = end - start + 1. \tag{3.3}$$

The normalized length also depends on the number if insertions ins and deletions del

$$normalLen = end - start + 1 - ins + del. \tag{3.4}$$

The normalized repeat length is defined such that, if divided by the unit length, it is the natural measure for the number of repeat units in the alignment. It can also be interpreted as the number of nucleotides in the tandem repeat of the subject sequence before insertions and deletions lead to its degeneration. This is illustrated in the following example.

Example:
The following imperfect tandem repeat with putative unit AAC aligned to its exact counterpart

```
AACAACAACGGAAC-ACAACAAC
||||||||||||||||||||||
AACAACAAC--AACAACAACAAC
```

has a length of 22 bp. The normalized length is 21 bp, which reflects the fact that we apparently found *complete* 7 units and that the repeat length would be 21 bp if not the one putative deletion and the two putative insertions had occurred.

3.5.2 Repeat unit format

The repeat unit can be printed in three different modes:

As is: The unit is printed as it appears in the first complete unit. E.g. the unit of an $(TCG)_n$ tandem repeat is TCG.

normalized (cyclic): The unit is printed in a normalized form, where among all cyclic permutations of the repeat unit the minimal alphabetic representation is chosen. In this mode, the $(TCG)_n$ repeat has CGT as its unit.

normalized (cyclic)+reverse complement: The unit is printed in a normalized form, where among all cyclic permutations of the repeat unit and its reverse complement the minimal alphabetic representation is chosen. In this mode, the $(TCG)_n$ repeat has ACG as its unit.

3.5.3 Print mode for repeat sequence

The repeat sequence of a tandem repeat can be printed after the general repeat information. There are three modes to print the repeat sequence:

don't print sequence: Repeat sequence is not printed.

print repeat sequence: Print the repeat sequence from start to end position in the subject sequence.

print repeat alignment: Print the alignment of the repeat sequence against its perfect counterpart.

3.5.4 Flanking nucleotides

Together with the repeat sequence, Phobos can print a specified number of nucleotides flanking the repeat. This is possible in all three sequence print modes. The flanking nucleotides are separated by a ":" from the repeat sequence.

3.5.5 Treating N's when computing the percentage perfection

If N's are treated as missense (mismatch or gap) in the alignment, they will also be treated as such when computing the percentage perfection. In this case the percentage perfection is computed according to

$$p = 100\,\frac{normalLen - mis - del - ins}{normalLen}. \tag{3.5}$$

where $normalLen$ is given in equation (3.4) above and mis, del, ins are, respectively, the number of mismatches, deletions and insertions.
If N's are treated as neutral in the alignment with a score of zero, the user can choose how they should be treated when computing the percentage perfection. The three possibilities are:

as mismatch: All N's are treated as being mismatches leading to a percentage perfection of

$$p = 100\,\frac{normalLen - mis - del - ins - Ns}{normalLen}. \tag{3.6}$$

as neutral: All N's are treated as neutral leading to a percentage perfection of

$$p = 100\,\frac{normalLen - mis - del - ins - Ns}{normalLen - Ns}. \tag{3.7}$$

as matches: All N's are treated as matches leading to a percentage perfection of

$$p = 100\,\frac{normalLen - mis - del - ins}{normalLen}. \tag{3.8}$$

Chapter 4

Examples

4.1 Alternative alignments

Let us assume the following content in the FASTA input file example.fas:

```
>Example 1
TT ACACAC ACACAG ACACAG ACACAC ACACAG ACACAG ACACAT ACACAC TT
```

The command to start Phobos from the command-line with a mismatch and gap penalty of -4 and using default values for all other parameters is

```
phobos -m -4 -g -4 ../data/examples.fas
```

This command yields the following output:

```
# Results computed with:
#    Phobos, version 3.2.5
# Parameter settings used in this search:
# Input file name:                  ../data/examples.fas
#    Sequence range to process:     1 - last sequence
#    Satellites searched for:       imperfect repeats
#    Recursion depth:               5
#    Minimum unit length:           1
#    Maximum unit length:           6
#    Minimum score:                 maximum (6, 0+1*unitlength )
#    Minimum length:                maximum (1, 0+0*unitlength )
#    Treat N's as:                  neutral
#    Max. succesive N's allowed:    4
#    Mismatch score:                -4
#    Indel score:                   -4
#    Minimum perfection:            0 %
#    Number of flanking nucletides: 0
#    Remove mostly overlapping satellites: yes
#    Write file with masked seq.:   no
#    Computing perfections treat N's: asMismatch
#    Report units as:               alphabetical normal form using reverse complement
#    Output format:                 extended phobos format
#    Print sequence mode:           print alignment
#
# Columns in lines of output providing repeat information:
# Column  1: Repeat unit length
# Column  2: Position at which repeat starts in current sequence
# Column  3: Position at which repeat ends in current sequence
# Column  4: Length of repeat in original sequence, i.e. (end-start+1)
# Column  5: Normalized repeat length, i.e. (end-start+1-insertions+deletions)
# Column  6: Repeat score
# Column  7: Repeat perfection
# Column  8: Number of mismatches
# Column  9: Number of insertions
# Column 10: Number of deletions
# Column 11: Number of N's
# Column 12: Repeat unit
#
>Example 1
sequence length: 52
  hexanucleotide         3 :        49 |     47 bp |     47 BP |     26 pt |  93.617 % |   3 mis |   0 ins |   0 del |   0 N |  unit ACACAG
Missense: (6,M),(24,M),(42,M),
ACACACACACAGACACAGACACACACACAGACACAGACACATACACA
||||| ||||||||||||||| ||||||||||||||| |||||
ACACAGACACAGACACAGACACAGACACAGACACAGACACAGACACA
## Finished successfully. ##
```

Among the two competing repeat units AC and ACACAG, Phobos has chosen the one with the higher alignment score.
In order to show all alignments with alternative repeat units the option “don’t remove mostly overlapping” has to be used. Starting Phobos with this option from the command-line

```
phobos -m -4 -g -4 ../data/examples.fas --dontRemoveMostlyOverlapping
```

yields the following output:

```
# Results computed with:
#    Phobos, version 3.2.5
# Parameter settings used in this search:
# Input file name:                    ../data/examples.fas
#    Sequence range to process:       1 - last sequence
#    Satellites searched for:         imperfect repeats
#    Recursion depth:                 5
#    Minimum unit length:             1
#    Maximum unit length:             6
#    Minimum score:                   maximum (6, 0+1*unitlength )
#    Minimum length:                  maximum (1, 0+0*unitlength )
#    Treat N’s as:                    neutral
#    Max. succesive N’s allowed:      4
#    Mismatch score:                  -4
#    Indel score:                     -4
#    Minimum perfection:              0 %
#    Number of flanking nucletides:   0
#    Remove mostly overlapping satellites: no
#    Write file with masked seq.:     no
#    Computing perfections treat N’s: asMismatch
#    Report units as:                 alphabetical normal form using reverse complement
#    Output format:                   extended phobos format
#    Print sequence mode:             print alignment
#
# Columns in lines of output providing repeat information:
# Column  1: Repeat unit length
# Column  2: Position at which repeat starts in current sequence
# Column  3: Position at which repeat ends in current sequence
# Column  4: Length of repeat in original sequence, i.e. (end-start+1)
# Column  5: Normalized repeat length, i.e. (end-start+1-insertions+deletions)
# Column  6: Repeat score
# Column  7: Repeat perfection
# Column  8: Number of mismatches
# Column  9: Number of insertions
# Column 10: Number of deletions
# Column 11: Number of N’s
# Column 12: Repeat unit
#
>Example 1
sequence length: 52
  hexanucleotide         3 :        49 |     47 bp |     47 BP |     26 pt |  93.617 % |   3 mi
s |   0 ins |   0 del |   0 N |  unit ACACAG
Missense: (6,M),(24,M),(42,M),
ACACACACACAGACACAGACACACACACAGACACAGACACATACACA
||||| ||||||||||||||||| ||||||||||||||||| |||||
ACACAGACACAGACACAGACACAGACACAGACACAGACACAGACACA
    dinucleotide         3 :        50 |     48 bp |     48 BP |     21 pt |  89.583 % |   5 mi
s |   0 ins |   0 del |   0 N |  unit AC
Missense: (12,M),(18,M),(30,M),(36,M),(42,M),
ACACACACACAGACACAGACACACACACAGACACAGACACATACACAC
|||||||||||| ||||| |||||||||||| ||||| ||||| ||||||
ACACACACACACACACACACACACACACACACACACACACACACACAC
## Finished successfully. ##
```

This time, Phobos reports both competing repeat units.

For a large number of alternative and competing repeat units, this output mode still provides too much information. In future versions of Phobos, it will be possible to specify the number of alternative and competing repeat units that shall be reported.

Chapter 5

Troubleshooting

5.1 Problem: Phobos has written the output header to the output file but nothing else

If Phobos is still running, it is currently processing the first sequence of the input file. Output is produced for each input sequence only after the analysis of this input sequence has been completed, which, depending the system resources and search parameters can take a while.

5.2 Problem: Phobos is running slower than expected

There are several reasons for Phobos to become slow in processing sequences and producing output:

Memory consumption: Depending on the length of the input sequence and the number of repeats therein, Phobos requires a certain amount of computer RAM. An analysis of the human genome, e.g., with default parameters requires 2 GByte of RAM in an *perfect* and *imperfect* search mode and 4 GByte in the *extend exact* search mode. Smaller genomes can even require more RAM if the number of tandem repeats is high. Memory consumption increases if the "minimum length of repeats" parameter is set to small values (which increases the number of repeats Phobos stores in memory), if the search includes mononucleotide repeats, and its particularly high in the "extend exact" search mode in which many tandem repeat seeds are stored. A lack of computer memory will cause Phobos to stall during the search.

Search parameters: A large value for the upper pattern size limit (> 10 in imperfect search) or a small absolute value of the gap and mismatch penalty (< 5) can cause Phobos to be slow. On larger data sets it is recommended to conduct a test search with a smaller pattern size range before starting a longer analysis with a larger pattern size range. It should be noted that analyses with the same search parameters of different sequences of the same length can require significantly different memory and CPU-time depending on the number of tandem repeats in the sequences.

This section is a compilation of problems Phobos users have come across. Please feel free to contact me by email if you experience problems with Phobos for which a solution could not be found in this section.