Nucleics

# UniSelect

## Users manual

**Version 1.1**

August 2002

# Table of contents

## Overview

**uniselect [options][-if infile|-id indir ][-of outfile|-od outpath]**

The UniSelect software is intended to automate the selection of E and T primer pairs from the UniSeq library for a given target sequence. The user can adjust the behavior of UniSelect by either modifying the initialization file or by specifying parameters on the command line. The output is a space delimited plain text file that can be used to robotically or manually pipette the optimal E and T primers for each UniSeq reaction.

## Introduction

The UniSelect program reads in user defined template sequences in the form of PHD files as generated by the PHRED base-calling software. PHD files contain the template sequence and the associated quality scores that are required to select the optimal E and T primer pairs.

Each template sequence is analyzed to ensure the template is of sufficient quality for primer pair selection. The minimum template quality score used by UniSelect can be set by the user allowing poor quality sequence to be excluded from further analysis.

Template sequences of sufficient quality are then searched for potential UniSeq primer binding sites. At each potential site, the E and T primer pair is analysed for template secondary structure with sites found within regions of stable secondary structure considered of lower fitness than those without secondary structure. The greater the calculated free energy of a secondary structure element, the progressively worse the fitness of the UniSeq primer site is weighted.

For each possible priming site passing the above checks, the position relative to the end of the sequence is fitness scored. The priming sites that are the closest to the end of the target sequence, and fulfill all other user adjustable parameters, are considered the most fit. The initial fitness score is then adjusted by consideration of local secondary structure and binding energies to produce the final fitness score. The primer with the largest fitness score is then selected as the optimal solution. In addition, UniSelect allows the user to request other primer solutions other than the optimal pair.

## Input and Output

### 1.1. File naming conventions

When using UniSelect and managing large projects with many sequence files, it is recommended that a user adopt a consistent convention for naming files and organizing them into directories (Example 1).

Nucleics recommends the convention that the sequences derived (i.e. PHD files) from each 96 or 384 well clone plate are stored together in one directory. Each sequence file should follow the St. Louis naming convention as used by the PHRAP assembly program. Users are encouraged to include the tray and well information in the template name by following a consistent naming convention to allows better integration with robotic pipetting systems. While the naming format can be user customized by modifying the regular expression used to extract the information (see *Regex Well*), Nucleics recommends following the St. Louis naming convention.

The St. Louis convention for identifying primer walks **`[string_integer]`** can also be used to selectively filter input files so that UniSelect will only process those clones intended for the next round of walking (see *Walk Depth*).

If clones need to be re-arraying to new plates, it is recommended that the file name be modified to include the new location appended to the beginning of the old name (Example 1). This format allows the history of each clone to be followed.

***Example 1. UniSelect naming convention***

*For a plate named T001, the following file and directories structure is recommended:*

```
T001_
     |- T001a01.b1.PHD
     |- T001a02.g1.PHD
     |- T001b12.g1.PHD
```

*If the template in wells a02 and b12 of plate T001 were re-arrayed to wells a01 and a02 of a new plate named T012 the following file and directory structure is recommended:*

```
T012_
     |- T012a01-T001a02.g1.PHD
     |- T012a02-T001b12.g1.PHD
```

### 1.2. Single file mode `-if [string]  -of [string]`

UniSelect can process PHD files in either single file or batch mode.

In single file mode UniSelect processes a single PHD file, suggests the optimal primer pair set, if one exists, and terminates. The default output in this mode is to the shell. The user can alternatively specify an output file for the results using **`-of`** option (Example 2).

***Example 2. Single file output***

```
> uniselect -if T001a10.phd
```

*output printed to the shell*

```
> uniselect -if T001a10.phd -of out.txt
```

*output written to the file* **`out.txt`**

### 1.3. Batch file mode `-id [string]  -od [string]`

In batch mode, UniSelect scans the supplied directory(s) and processes all PHD files found (see *Walk Selection* and *Suffix*). The default output is to a file in the current working directory named

after the input directory with the suffix **.uni**. This output file contains a suggested primer pair for each processed PHD file for which a primer pair can be found.

A user can change the output path using the **-od** option, however, UniSelect does not allow the user to name individual output files in batch mode (Example 3).

***Example 3. Batch mode output***

```
> uniselect -id data
```

*output written to the file* **data.uni**

*The output path can be changes using the* **–od** *argument*

```
> uniselect -id data -od outdir
```

*output written to the file* **outdir/data.uni**

### 1.4. Directory recursion -r

In batch mode, all the directories beneath the directory specified on the command line can be recursively scanned for PHD files using the **-r** option. For each directory below the specified directory, the PHD files will be processed as a set and an output file produced (Example 4). Each output file named is based on the directory containing the files.

Recursive runs can result in the processing of large numbers of directories with a consequential large number of output files being produced. This raises the possibility that two output files will possess identical names. To avoid this potential problem UniSelect can use long non-clashing names (see *Non-Clashing*) for the output files. Non-clashing names use the directories' local path to create a unique name (Example 4). The character or string used to separate directory names in the non-clashing output name can be modified by the user.

***Example 4. Recursive processing of directories***

*For a directory structure such as:*

```
data_
      |- T001
      |- T002_
            |- T002_1
      |- T003
```

```
> uniselect -r -id /usr/data
```

*Produces the output files:*

```
data.uni
T001.uni
T002.uni
T002_1.uni
T003.uni
```

```
> uniselect -r --non-clashing -id /usr/data
```

*Produces the output files:*

```
data.uni
data.tray01.uni
data.tray02.uni
data.tray02.tray02_1.uni
data.tray03.uni
```

*In contrast, non-recursive processing of the same directory structure:*

```
> uniselect –id /usr/data
```

*produces*

```
data.uni
```

*containing the results of processing the files contained immediately within*

```
/usr/data/
```

### 1.5. Initialization file `-i [*.init]`

All parameters and options used by UniSelect in processing files and selecting primer pairs can be set either from the command line, or from within the initialization file. The default name of the initialization file is **`uniseq.init`**. Settings given on the command line override the settings contained within the initialization file.

The initialization file is in text format and contains a complete listing of both the EO and the TO primers, as well as all user modifiable parameters and options (Example 5). The use of an

initialization file avoids the need for the user to pass larger numbers of arguments via the command line. If this file is lost or damaged, UniSelect will prompt the user to create a new initialization file with the options and parameters set to their defaults values.

***Example 5. Default user adjustable entries found in the initialization file***

```
[overlap] 20
[good_sequence] 20
[regex_well] (.*)([[:alpha:]]{1})([[:digit:]]{1,2})[\._]
[suffix] phd
[recurse] 0
[clobber] 0
[nonclashing] 0
[ignore_names] 0
[fasta] 0
[384well] 0
[solution] 1
[verbose] 0
[separator] .
[test] 0
[walkdepth] 0
```

The use of an alternative initialization file can be specified on the command line using the **-i** option (Example 6).

***Example 6. Use of a custom initialization file***

```
> uniselect -i custom.init -id /usr/data
```

## Template Analysis

UniSelect identifies the optimal primer pair using both the quality scores of the template sequence and the presence of any secondary structure elements that will interfere with the primer binding to the template. The user can modify the following two parameters.

### 1.6. Good Sequence `--good-sequence [integer]`

*Good Sequence* defines the PHRED quality level at which sequence data is considered reliable. Used in conjunction with *Overlap*, a user can define the minimum length of overlapping regions of reliable sequence between successive primer walks. Overlaps of reliable sequence are helpful in assembling sequence data using programs such as PHRAP. The default value is 20.

### 1.7. Overlap `--overlap [integer]`

Used in conjunction with *Good Sequence*, the *Overlap* parameter is used to define the minimum required length of overlapping reliable sequence data between successive primer walks. The value specified by a user only sets a minimum acceptable overlap length in number of bases. The default is 20 bases.

## General Options

UniSelect has the following general options.

### 1.8. Verbose `-v`

*Verbose* sets UniSelect to verbose output. Much internal detail is included for each template analysis. The resulting output file can be quite long.

### 1.9. Suffix `--suffix [string]`

*Suffix* allows the user to change the file name suffix used to filter input files in batch mode. The suffix may be set to any user-defined string. The default is **`.phd`**.

### 1.10. Regex Well `--regex [regular expression]`

*Regex Well* permits the user to define the regular expression used to extract the template DNA tray and well location from the file name. *Regex Well* expressions obey the POSIX regular expression convention (for further information on regular expressions see the GNU Regex

manual). The user must specify three sub patterns when defining the pattern: the tray name, the row letter, and the column number (Example 7).

***Example 7. Regex template extraction***

```
(.*)([[:alpha:]]{1})([[:digit:]]{1,2})[\._]
```

*Here sub-pattern 1* **(.*)** *matches any leading string. A single letter is matched by sub-pattern 2* **([[:alpha:]]{1})** *and sub-pattern 3* **([[:digit:]]{1,2})** *matches 1-2 consecutive digits. All three sub-patterns must follow in consecutive order to be a match. Finally* **[\._]** *marks the end of the section of the file name which UniSelect will inspect for the necessary information. In this example either character* **.** *or* **_** *mark the end of the section.*

*Using the above pattern, Regex Well would extract from a file with the name* **T001a10.phd**

Sub-pattern 1: **T001**
Sub-pattern 2: **a**
Sub-pattern 3: **10**

However a user defines their regular expression care must be taken to create names that follow the expected pattern.

### 1.11. Ignore Names `--ignore-names`

*Ignore Names* stops UniSelect from attempting to extract tray and well information from the file names. This option can prove useful when the files do not follow a consistent naming format.

### 1.12. Non Clashing `--non-clashing`

The use of long non-clashing output file names is specified by the *Non Clashing* option. Non-clashing names are based on the complete local path to the current directory to be processed. The path separators (e.g. **/** ) are replaced with a user specified separating string (see *Separator*). A path such as **data/bac1/T001** with the separator **'.'** (period) would generate the name **data.bac1.T001.uni**.

### 1.13. Separator `--separator [string]`

The *Separator* option allows the user to select the separator used in concatenating directory names in long non-clashing names. The default is **'.'** (period).

### 1.14. Clobber `--clobber`

*Clobber* permits UniSelect to overwrite existing output files. The default is to not overwrite output files.

### 1.15. Fasta `--fasta`

*Fasta* informs UniSelect that the input file is in FASTA format. No quality score is required as the sequence is assumed to contain no errors. This option is not recommended for templates with regions of poor quality, such a sequence trace files.

### 1.16. 384 Well `--384-well`

The use of 384-well PCR plates is set using *384 Well*. The default option is 96 wells.

### 1.17. Solution `--solution [number]`

The *Solution* option allows the user to select primer pairs other than the optimal. A user can request any solution from 1 to 5 provided the solution exists. If the requested solution does not exist then UniSelect reports no solution has been found. The default is to provide the first solution (i.e. the optimal choice).

### 1.18. Test `--test`

*Test* prevents UniSelect from recording the primer usage in the initialization file. This can be useful for performing *in silico* experiments with program settings without affecting the usage statistics.

### 1.19. Walk Depth `--walk-depth [number]`

*Walk Depth* allows the user to filter sequence files based on the walk depth according to the St. Louis naming convention (Example 8).

***Example 8. First and second walk files on template T100a10***

```
T001a10.phd
T001a10_1.phd
T001a10_2.phd
```

### 1.20. Biomek2000 `--bm2k`

This switch causes UniSelect to produce an output file that can be immediately used with our Biomek2000 tcl script without editing. The standard output is more user friendly and contains additional information not required by the robot script. If a user has requested a verbose mode (`-v`, `-vv`) then the output file will need to be edited prior to use by the robot.

The robot expects the format of the file to be:

Line 1 contains the number of jobs following below, each subsequent line contains three values which define a job, the destination well, the Eo index, and To index.

```
Line1:   Number_of_jobs

Line2:   Destination_well     Eo_index  To_index
```

## Technical support

Nucleics provides expert technical support for the UniSeq system.

Please address any questions or comments to:

uniseq@nucleics.com