

Chapter 2

Probe Design for Expression Arrays Using OligoWiz

Rasmus Wernersson

Abstract

Since all measurements from a DNA microarray is dependant on the probes used, a good choice of probes is of vital importance when designing custom microarrays. This chapter describes how to design expression arrays using the *OligoWiz* software suite. The desired general features of good probes and the issues which probe design must address are introduced and a *conceptual* (rather than mathematical) description of how OligoWiz scores the quality of the potential probes is presented. This is followed by a detailed step-by-step guide to designing expression arrays with OligoWiz.

The scope of this chapter is exclusively on expression arrays. For an in-depth review of the entire field of probe design (including a comparison of different probe design packages) as well as instructions on how to produce special purpose arrays (e.g., splice detection arrays), please refer to (1).

Key words: Probe design, probe selection, expression array, oligonucleotide array, DNA microarray, software, bioinformatics, transcripts.

1. Introduction

A good choice of probes is vital to the usefulness of a microarray since the probes determine what signal will be detected (from both intended and non-intended targets). In summary a good probe must fulfill the following criteria:

- An ideal probe must discriminate well between its intended target and all other potential targets in the target pool.
- The probe must be able to detect concentration differences under the applied hybridization conditions.

OligoWiz website: <http://www.cbs.dtu.dk/services/OligoWiz/>

These two points are the ultimate goal to achieve for all probe design software packages, even if the actual algorithms used can be quite different (1). The following sections describe how this is handled in the OligoWiz software package.

1.1. Introducing OligoWiz

Since the computational burden of performing the scoring of all possible probe positions is substantial, OligoWiz has been implemented as a *client-server* solution.

The workflow is as follows: The user interfaces with the Graphical User Interface (the “client” – written in Java for platform-independent use, see Fig. 2.1), and selects a dataset and a set of parameters for the probe design project. Next the

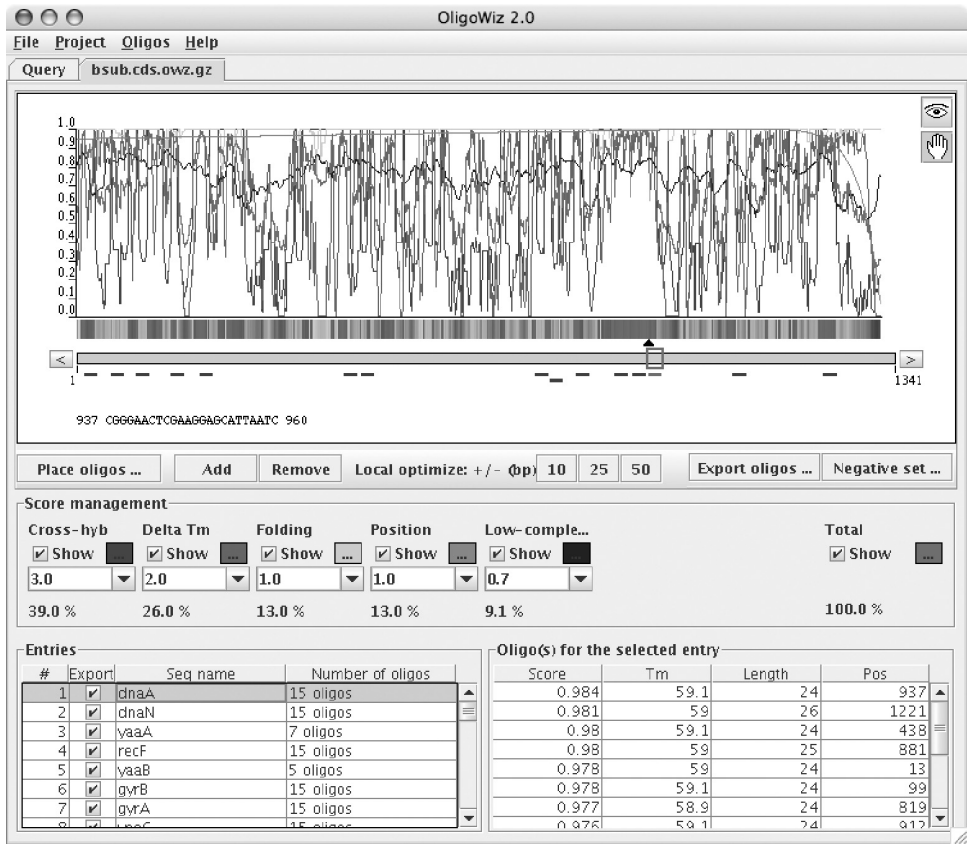


Fig. 2.1. **OligoWiz 2.0 screenshot.** This screenshot shows the main functionality of the software – including the graphical representation of the probe-“goodness” scores and the placement of probes along the selected transcript. The orange bar below the curves represents the currently selected transcript (dnaA). In this example a short-mer (24–26 bp) probe design for *Bacillus subtilis* is in progress, and up to 15 probes per transcript have been placed. The probes are visualized as lines below the transcript, and details are provided in the lower right-hand corner. Please note that the five scores are color-coded (cannot be seen here) – examples of the color coding is found at the OligoWiz website and in OligoWiz publications (1–3).

data is uploaded to the server (hosted at a multi-CPU super-computer located at the Center for Biological Sequence Analysis at the Technical University of Denmark) where all the computationally heavy algorithmic processing takes place. Once calculation of a particular dataset is completed a datafile with scoring information about *each potential* probe along all transcripts in the dataset is returned to the user, and all further work on the actual probe selection happens in a completely off-line fashion using the GUI.

1.2. Probe Suitability Scores in OligoWiz

OligoWiz uses a scoring-scheme that works as follows: For each position along all transcripts in the input dataset the suitability of placing a probe here is evaluated according to five criteria: Cross-hybridization, ΔT_m , Folding (self-annealing), Position (within the transcript) and "Low-complexity." Each individual score has a value between **0.0** (not suited – a bad position for placing a probe) to **1.0** (well suited – no problems detected). The individual scores are then combined with different weights (e.g., Cross-Hybridization is more important than Low-Complexity, *see Fig. 2.1* for the default values) to form a **Total score** which is also normalized to be between **0.0** and **1.0**. The actual selection of the best position for probe placement is based on the Total score.

In the following sections the *conceptual* workings of the individual scores will be described. The actual formulas for the calculations are found in the two main OligoWiz publications (2, 3).

1.2.1. Cross-Hybridization

As mentioned previously a vital property for a probe is to pick up only the intended signal. A way to ensure this is to avoid probes that may hybridize (partially) to other transcripts. It has been shown (4) that a 50-mer will detect a significantly false signal from an unintended target that has more than 75–80% identity at the sequence level. Also, short stretches (> 15 bp) of complete complementarity will give rise to a signal from cross-hybridization. Similar result for short oligos (23–27 bp) has recently been shown by (1).

The perfect way to get around this problem is to calculate the actual hybridization energy between all probes and all targets at the correct individual concentrations. However, since the concentrations of the targets are not known, and since such calculations are very time-consuming we have opted for an approximate solution: screen the entire genome (for prokaryotes and small eukaryotes) or transcriptome (Unigene collection for large eukaryotes, like mammals) using BLAST (5, 6) for regions with substantial similarity to the transcripts in question. By default regions with more than 75% similarity over at least 15 bp is considered to be problematic.

1.2.2. ΔT_m

Another important aspect of probe design is to ensure uniform hybridization conditions throughout the array. Traditionally this has been done by controlling the GC ratio within the probe. OligoWiz addresses this issue by forcing the distribution of T_m (melting temperature) to be as narrow as possible.

This is done in two ways:

- A " ΔT_m " score¹ that evaluates how far the T_m of a potential probe is from the mean T_m of all potential probes.
- Allowing the length of the probes to vary. **Fig. 2.2** shows how the T_m distribution of a set of oligonucleotides becomes increasingly narrow, if the most optimal length (within an interval) can be chosen. Working with short probes it is the experience of the OligoWiz authors that even allowing the length to vary just between 24–26 bp will improve the T_m profile. *Finding the optimal length is the very first step performed by OligoWiz:* For each position the most optimal length within the user-specified interval is determined, and this length is used for the calculation of all other scores.

1.2.3. Folding

To ensure uniform hybridization conditions for all probes on the microarray, the probes should avoid self-annealing (folding). The classical way of investigating this issue is to calculate the free

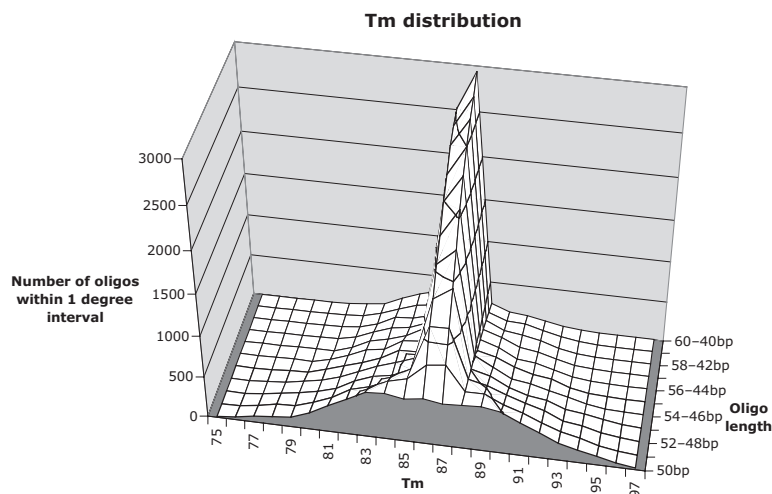


Fig. 2.2. T_m distribution in optimized length intervals of oligonucleotides. This figure shows how the T_m -distribution of a large set of oligonucleotides (based on all 50 mers within the Yeast genome) can be made increasingly narrow by allowing the length to vary and selecting the most optimal length within each interval. (Based on data from (2)).

¹ Listed as "Delta- T_m " in the interface.

energy of potential secondary structures using programs such as MFOLD (7). However, using MFOLD is very time consuming² and for this reason approximate methods that are two orders of magnitude faster was developed for OligoWiz (3). Briefly, this method is based on the idea of aligning the oligo to itself using a dinucleotide alphabet using dynamic programming (8) and a substitution matrix based on the dinucleotide binding energies. The resulting alignment will represent the lowest folding energy state given the input sequence. This approximate method is in good agreement with MFOLD (*see* (3), **Fig. 2.2**) – especially for the sequences with strong secondary structure, which are the most important to avoid when designing probes for DNA microarray. Since all possible probe positions along all target transcripts must be scored, the calculations can be done in a sliding window fashion, where most of the dynamic programming matrix from the previous position can be reused, this contributes significantly to the speed-up. Please *see* (3) for further details on the implementation.

1.2.4. Low Complexity

In order to avoid picking up background signal, probes that contain a lot of "sub-words" that are common in the genome/transcriptome should be avoided. This can be illustrated with the following example (human DNA):

Oligo with low-complexity:

AAAAAAGGAGTTTTTTTCAAAAACTTTTTAAAAAGCTTTAGGTTTTTA

Oligo without low-complexity:

CGTGACTGACAGCTGACTGCTAGCCATGCAACGTCATAGTACGATGACT

In OligoWiz, this problem is addressed by counting the occurrence of all 8 bp "words" in the genome/transcriptome and scoring the degree to which a probe consist of frequent sub-words.

1.2.5. Position

The optimal position within the transcript for placing a probe depends on the labeling and/or amplification method used. When using standard poly-T priming (targeting the poly-A tail of eukaryotic transcripts) the labeling starts from the 3' end of the transcript. Since there is a certain probability that the reverse transcriptase will not complete the synthesis of cDNA in full length, most signals are detected using probes targeting the 3' end of the transcript. In OligoWiz the following position preference models are built in.

² 2 seconds for a 30-mer and 16 minutes for all 30-mers in a 500 bp transcript at OligoWiz reference platform at the time.

- **Poly-T priming:** Push probes towards the 3' prime end (Probabilistic model of the labeling from the 3' prime end).
- **Random priming:** Avoid probes at the extreme 3' prime end (Probabilistic model of the labeling using random hexamers).
- **Linear 5' preference:** 1.0 at the 5' end and decreases linearly to 0.0 over 2000 bp.
- **Linear 3' preference:** As the 5' preference, but counting from the 3' end instead.
- **Linear mid preference:** 1.0 at the midpoint decreasing to 0.0 over 1000 bp to each side.

Observe that it is possible to completely ignore the position score, by setting its weight to **0.0**. This is especially useful in situations like placing splice-junction probes, where the position is constrained by the gene structure.

1.3. Rule Based Placement of Probes

As mentioned previously, the OligoWiz server returns a datafile to the client (the graphical interface) which contains *scoring of all possible probes*. At this point no decisions about the actual placement (how many per transcript, spacing etc.) of the probes have been made. All the computations on the placement of the probes is performed solely on the user's own computer in a completely off-line manner. This means that once the data file has been created it contains everything needed for further work, and can be stored on the user's own computer/network or be shared with collaborators using email, for instance.

The actual placement of the probes is done using a rule based method (*see Fig. 2.3* for an overview of the options). The placement algorithm is as follows (repeated for each transcript):

1. **Apply filters:** If any filters have been defined (e.g., requiring the total-score to be above a certain value), start by masking out the regions disallowed by the filters. For the advanced optional use of filter please *see (1)*.
2. **Place probe:** Select the currently available position with the highest Total score for probe placement.
3. **Mask out surrounding positions:** Positions within the desired minimum distance are masked out.
4. **Repeat/terminate:** Terminate if the maximum total number of probes has been reached or if no more positions are available. Otherwise, go to step 2.

Since the computationally heavy calculations (scoring of all probe position) have already been performed on the server, the placement algorithm is fast. This makes it possible to

Oligo placement [bsub.cds.owz.gz]

General oligo placement

25 : Min distance between oligos (bp)

15 : Max number of oligos/sequence

☐ Unlimited

0.0 : Minimum total score allowed

Replacement behavior

☒ **Replace existing oligos**
Existing oligos are discarded and have no effect on the placement of new oligos

☐ **Keep existing oligos**
Notice: Existing oligos are considered during the search.

Filters

☒ **Only consider regions annotated as exons**
Unselect this option to enable custom filters

Custom filters: PERL style Regular Expressions are supported

Region Include: \ (?E+ \) ?
Allow only oligos in regions which match the expression

Region Exclude:
Exclude oligos in regions which match the expression

Oligo Include :
Individual oligos must match the expression

Oligo Exclude :
Exclude oligos matching the expression

☐ Search sequence ☒ Ignore case

☒ Search annotation

Apply Apply to all Help ... Dismiss

Fig. 2.3. **Probe selection dialog.** The spacing criteria are specified in the topmost box. The use of filters and sequence feature annotation (e.g., intron/exon structure) are not described here. For further details please refer to the OligoWiz website and (1).

experiment with the probe placement parameters, evaluate the result, and refine the parameter in a real-time iterative fashion.

1.4. Exporting the Probe Sequences

The final step in the probe design process will be to actually order the array (e.g., NimbleExpress) or the oligonucleotides to be spotted. In order to make this step easy, OligoWiz support exporting the probe sequence to both FASTA and TAB format, and has the option of reverse-complementing the probes (if needed) and automatically creating PM/MM probe pairs, if that is desired. Furthermore, it should be noted that a Material and Methods section describing the parameters used in the probe design is auto-generated and added to the file,

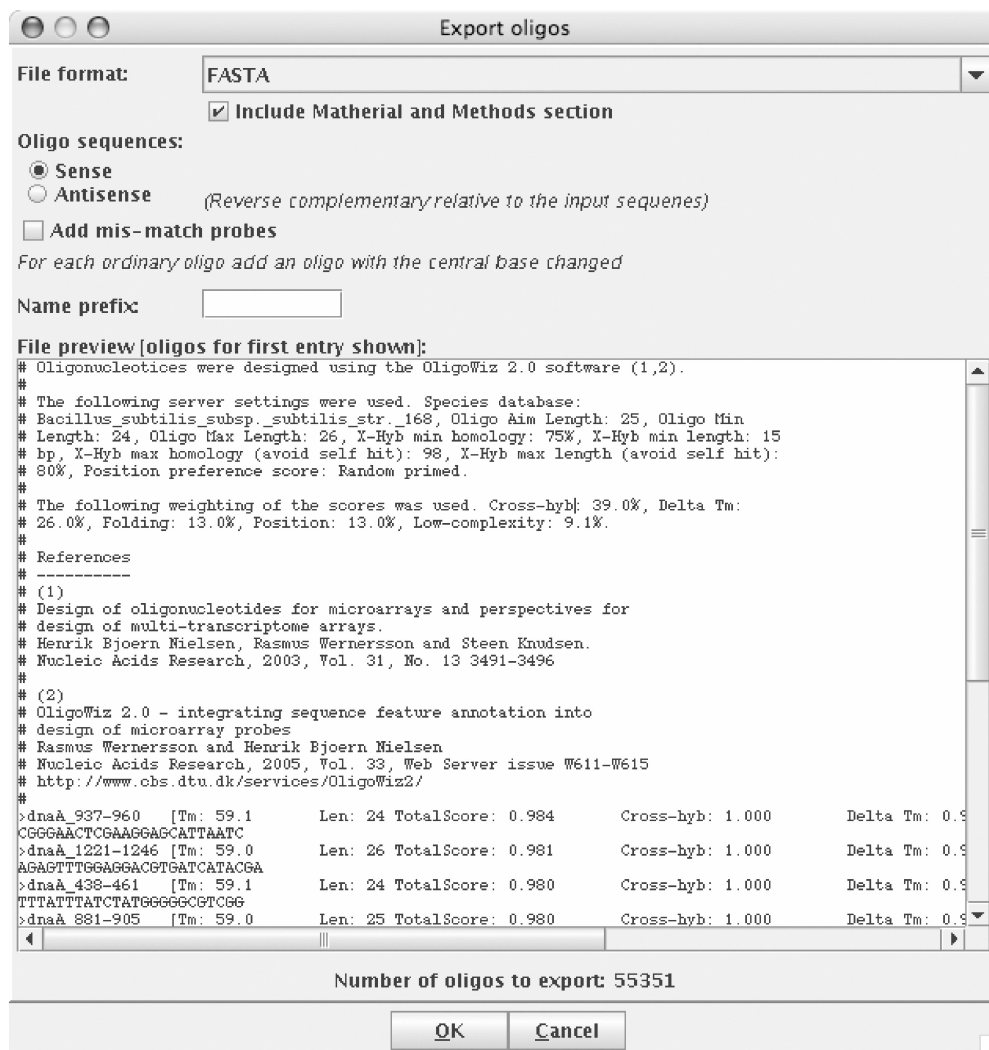


Fig. 2.4. OligoWiz probe sequence export options.

documenting the probe design process (see Fig. 2.4 and step 10 in the step-by-step guide).

2. Materials

An internet connected computer with Java 1.4 (or newer) installed. The OligoWiz client is tested on Windows, Mac OS X, Irix, Solaris and Linux – it is written with cross-platform use in mind and should work on virtually any operating system for which a Java Runtime Environment (JRE) exists. The optional

use of a local installation of the OligoWiz server software is not covered here, please *see* (1) and the OligoWiz website for further details.

3. Methods

This section summarizes the steps the user has to go through to select probes for an expression array.

1. **Prepare target sequences in FASTA format.** (For instructions on how to use TAB files please *see* (1, 9) – or the descriptions on the OligoWiz website). The very first step is to identify the sequences that the array should detect. This could for example be an entire prokaryotic genome or a set of transcripts from the human genome/transcriptome. For prokaryotic sequences a file prepared from the CDS (protein coding genes) regions of the full genomic sequence is recommended. In many cases a FASTA file with only the transcripts/CDSs can be downloaded from the same data-source as the full genomic builds. For higher eukaryotes (e.g., Human or Mouse), sequences from the UNIGENE collections are recommended. Observe that it is important to also include control targets/genes – since most normalization algorithms used in the downstream processing assumes that only a minor (10%) proportion of the transcript vary from array to array (10). *See Note 1* for further details about the input data.
2. **Launch the OligoWiz client.**
 - 2.1. Download the most recent version of the OligoWiz client from the OligoWiz website: www.cbs.dtu.dk/services/OligoWiz/.
 - 2.2. Download Java version 1.4 (or newer) if it is not already installed on the local computer. Instruction on how to do this on various platforms (Windows/Linux/Mac) is detailed on the webpage.
 - 2.3. Launch the OligoWiz client by double-clicking on the JAR file (Windows and Mac) or from the command-line (Linux and UNIX). *See Note 2* for issues relating to the memory usage of the program.
3. **Select input file.** Click the “...” button next to the *Input FASTA or TAB file* field (*see Fig. 2.5*), and select the FASTA file prepared in Step 1. The OligoWiz client will suggest a unique filename for the result file (not generated yet) – accept this, or customize the filename/placement if desired.

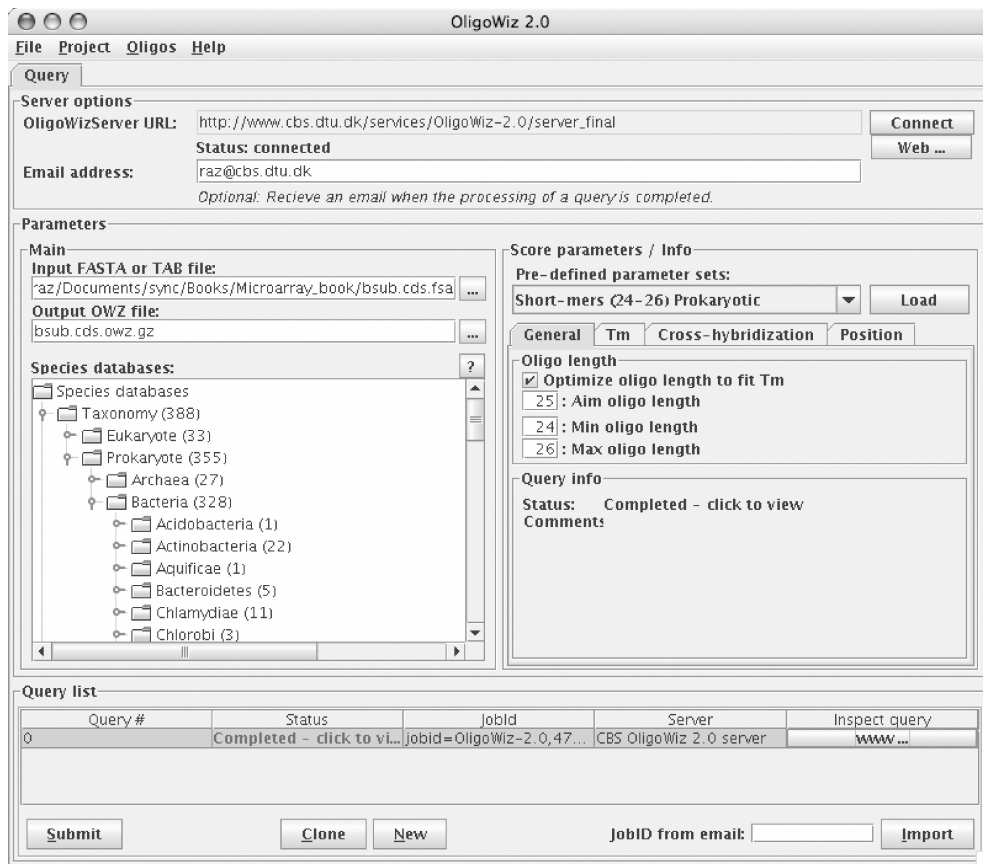


Fig. 2.5. OligoWiz query launch page.

4. **Select species database.** Select the species database that will be used for calculating the Cross-hybridization and Low-Complexity scores. A full description of all the databases³ is available on the OligoWiz website. (If the species-tree is empty, please refer to **Note 3** describing how to troubleshoot network issues).
5. **Customize score parameters.** Select the best fitting predefined parameter set in the *Score parameters/info* box and press *Load* (see Fig. 2.5). The predefined parameter sets can be customized further, as described below:
 - 5.1. **Oligo Length:** Determines if OligoWiz should aim at a fixed length or allow the length to vary within an interval in order to optimize T_m (recommended).
 - 5.2. **T_m**
 - 5.2.1. Select if OligoWiz should determine the optimal T_m (recommended) – alternatively a specific T_m to aim for can be specified.

- 5.2.2. Select if OligoWiz should use a DNA:DNA or RNA:DNA model for calculating the T_m . Select DNA:DNA if DNA is to be hybridized to the array and RNA:DNA if RNA is used (this is typically the situation).

5.3. Cross-Hybridization

- 5.3.1. Set the cut-off values of when a BLAST hit is to be considered: % minimum similarity and minimum length. Hits below this threshold will be completely ignored. It is recommended to use the default values.
- 5.3.2. Set the cut-off when a BLAST hit is considered a “self-hit” (the target sequence it self). For prokaryotic arrays the default values are recommended – if the input data is transcripts for a complex eukaryotic organism with a large degree of alternative splicing, the issue of detecting self-hits is more complicated. In this case it is recommended to lower the self-hit criteria. A pragmatic solution is to lower the self-hit length criteria to ~40% (0.4) – *see (1)* for a detailed discussion.
- 5.4. **Select position model.** For labeling protocols using poly-T (usually the case for running eukaryotic arrays) select the *Poly-T* option. For labeling protocols using random hexamers (usually the case for prokaryotic arrays) select the *Random priming* option.

6. Submit the query

- 6.1. **Optional step:** Enter your email address in the *Email address* field – this will make the server send you an email once the processing is completed with a link to direct download of the result data file. This is especially useful for long running queries.
- 6.2. **Press the “Submit” button**
7. **Wait for the server to finish processing the query.** The status of the processing can be seen in the *Query List* table. Once the processing has completed, the data file (file type: .owz.gz) will automatically be downloaded and stored on the local computer.
8. **Load the data file.** Double-click on the downloaded query in the “Query List” table to load in the data. This will load in the data and launch the main interface for placing probes (*see Fig. 2.1*).

Notice: If the data file has been downloaded manually by following the link in the server-generated email, the data can be loaded by using the File -> Open menu option.

9. Place probes

- 9.1. **Adjust score weights (if needed).** It is recommended to keep the default settings. However, notice that it's possible to disable a score by setting its weight to **0.0**.
- 9.2. **Bring up the Oligo Placement window.** Press the “*Place Oligos...*” button to launch the probe selection dialog (*see Fig. 2.3*).
- 9.3. **Select probe placement criteria.** For short probes (~25 bp) 8 probes or more per target sequence is recommended, for long probes (50–70 bp) 2–4 (or more) is recommended (*1*).
- 9.4. **Apply selection criteria.** Press the *Apply to all* button to search for probes fulfilling the criteria in the entire data set. (The *Apply* button can be used to test the criteria on a single sequence).
- 9.5. **Inspect the placement of the probes.** Keep the probe placement window open, and inspect the placement of the probes in the main window. Notice that both the *Entries* and *Oligos* lists can be sorted by clicking on the header elements. This makes it easy to identify target sequences for which no or few probes have been selected.
- 9.6. **Repeat step b-e if needed.**
10. **Export probe sequences.** Press the “*Export oligos...*” button to bring up the Probe Export window (*see Fig. 2.4*). The sequences can be exported in FASTA and TAB format. Optionally the probe sequences can be exported as anti-sense probes and/or pairs or PM/MM (perfect match/Mis-match) probes can be generated. In most cases the probes should be saved as “sense” probes in FASTA format – however, it is important to make sure that the strandness is correct for the protocol to be used in the lab.
11. **Optional: Export negative set.** If a sub-set of the target sequence proves to be difficult to design probes for, this sub-set can be extracted from the full set of target sequences, by pressing the *Export negative set* button. This makes it possible to isolate the troublesome cases, and re-run the entire probe-design process for these sequences only with more relaxed settings (or alternatively deciding NOT to target these sequences in the array design).

4. Notes



1. **Problems related to input data:** The most common source of problems with running OligoWiz is problems with the input data:

- 1.1. Please make sure that the data is in a supported file format (TAB or FASTA). Notice that the file must be a text-only file (an otherwise correctly formatted FASTA file within a MS-Word document will NOT work).
- 1.2. Please make sure that the file contains the sequences of the transcripts/genes which should be targeted. Submitting a file with a single large DNA sequence representing an entire prokaryotic genome will not work. OligoWiz is designed to work in a gene/transcript oriented way (for comments on how to design a chromosomal tiling array please *see (1)*).
- 1.3. Please make sure that the input sequences are of a sufficient length. Entries that are shorter than the minimum probe length will be discarded.
2. **Memory problems:** For very large datasets, the default amount of memory available to Java may become a problem. As a rule of thumb more memory may be needed if a FASTA file with more than 10,000 sequences (average prokaryotic CDS length) is submitted. The OligoWiz webpage contains detailed instruction of how to start the OligoWiz client with more memory on various platforms.
3. **Network problems:** If the OligoWiz client fails to connect to the OligoWiz server (the species database list remains empty, and the connection status remains “not connected”) it is most likely to be due to problems with the network setup. The OligoWiz client communicates with the server using HTTP (like a web browser), and it needs a direct connection rather than going through a HTTP proxy. If the local network setup uses a HTTP proxy (inspect the browser proxy settings – or ask the local system administrator), this is likely to be the cause of the problem. The OligoWiz website contains a description of a work-around of this issue.

References

1. Wernersson, R., Juncker, A.S. and Nielsen, H.B. (2007) Probe Selection for DNA Microarrays using OligoWiz. *Nature Protocols*, 2, 2677–2691.
2. Nielsen, H.B., Wernersson, R. and Knudsen, S. (2003) Design of oligonucleotides for microarrays and perspectives for design of multi-transcriptome arrays. *Nucleic Acids Res*, 31, 3491–3496.
3. Wernersson, R. and Nielsen, H.B. (2005) OligoWiz 2.0-integrating sequence feature annotation into the design of microarray probes. *Nucleic Acids Res*, 33, W611–W615.
4. Kane, M.D., Jatke, T.A., Stumpf, C.R., Lu, J., Thomas, J.D. and Madore, S.J. (2000) Assessment of the sensitivity and specificity of oligonucleotide (50 mer) microarrays. *Nucleic Acids Res*, 28, 4552–4557.
5. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool. *J Mol Biol*, 215, 403–410.

6. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25, 3389–3402.
7. Zuker, M. (1994) Prediction of RNA secondary structure by energy minimization. *Methods Mol Biol*, 25, 267–294.
8. Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48, 443–453.
9. Wernersson, R. (2005) FeatureExtract-extraction of sequence annotation made easy. *Nucleic Acids Res*, 33, W567–W569.
10. Workman, C., Jensen, L.J., Jarmer, H., Berka, R., Gautier, L., Nielsen, H.B., Saxild, H.-H., Nielsen, C., Brunak, S. and Knudsen, S. (2002) A new non-linear normalization method for reducing variability in DNA microarray experiments. *Genome Biol*, 3, research0048.

DNA Microarrays for Biomedical Research

Methods and Protocols

Dufva, M. (Ed.)

2009, XII, 252 p. 54 illus., 1 illus. in color., Hardcover

ISBN: 978-1-934115-69-5

A product of Humana Press