OMP Developer Edition (OMP DE) Documentation

(Revised, February 22, 2010)

DNA Software, Inc.
334 E. Washington Street
Ann Arbor, Michigan 48104
Phone: +1 734-222-9092
E-mail: support@dnasoftware.com
Website: www.dnasoftware.com

Table of Contents

Chapter 1. General Notes

1.1 How to Read this Document

Chapter 2 introduces the user to a variety of common OMP DE experiments, strategies to use them and information on keywords of particular interest. Chapter 3 and 4 describe all keywords associated with OMPServer and AOI2Server. Implemented keywords are written in blue font and are usually followed by a valid value.

Hints:

1) Although all Keywords for OMPServer and AOI2Server are listed, not all are necessary at the same time. Example files generated with the Visual OMP software package are the best source of valid input and output files.

2) Descriptions for the keywords use a valid value as an example for the keyword.

3) Simulations and Designs can be made very complex. When starting an experiment, use a couple of simple keywords first, run a successful experiment and then add more parameters.

1.2 How to Run an Experiment (Windows)

OMP DE for Windows is composed mainly of two engines: OMPServer.exe and AOI2Server.exe OMPServer is used for simulation and analysis. All secondary structures, suboptimal structures, numerical analysis and main thermodynamics are computed by this engine. AOI2Server is used for probe and/or primer design. OMPServer.exe can be run by command line, through a dialog box or using a COM interface. AOI2Server.exe is run only through the command line.

1.2.1 Running the Command Line Executables

Open a command prompt window:

On Windows 2000 or Windows XP:

1) In Windows 2000 or XP: Go to the Start Button and select “Run”.
2) A dialog box should appear
3) Type in without quotation marks “cmd” and press enter

On Windows Vista:

1) Hold down the shift-key and right-click on the desktop or in a folder and select “Open Command Window Here”

Run OMPServer or AOI2Server

1) Go to the directory containing OMPServer.exe and AOI2Server.exe (default is C:\Program Files\DNASoftware\OMP):

a) To go to the C-drive, type the following without quotes: “cd C:\”

b) To go to the default directory for OMPServer and AOI2Server, type the following without quotes: “cd Program Files\DNASoftware\OMP”

2) Type “OMPServer.exe/CMD Input.oef Output.oof” where Input.oef is any input file and Output.oof is any output file to run OMPServer. Or type “AOI2Server.exe/CMD Input.odf Output.osf” where Input.odf is any input file and Output.osf is any output file to run AOI2Server.

If the input and output files are located in a directory other than the one containing OMPServer and AOI2Server, their full paths need to be given in the run simulation / run design commands. Like for example: “OMPServer.exe/CMD C:\OMPData\input.oef C:\OMPData\output.oof” in case the input-file is saved in the C:\OMPData directory and the output-file needs to be saved in this same directory.

1.2.2 Running the Dialog Box

Open a command prompt window:

On Windows 2000 or Windows XP:

1) In Windows 2000 or XP: Go to the Start Button and select “Run”.
2) A dialog box should appear
3) Type in without quotation marks “cmd” and press enter

On Windows Vista:

1) Hold down the shift-key and right-click on the desktop or in a folder and select “Open Command Window Here”

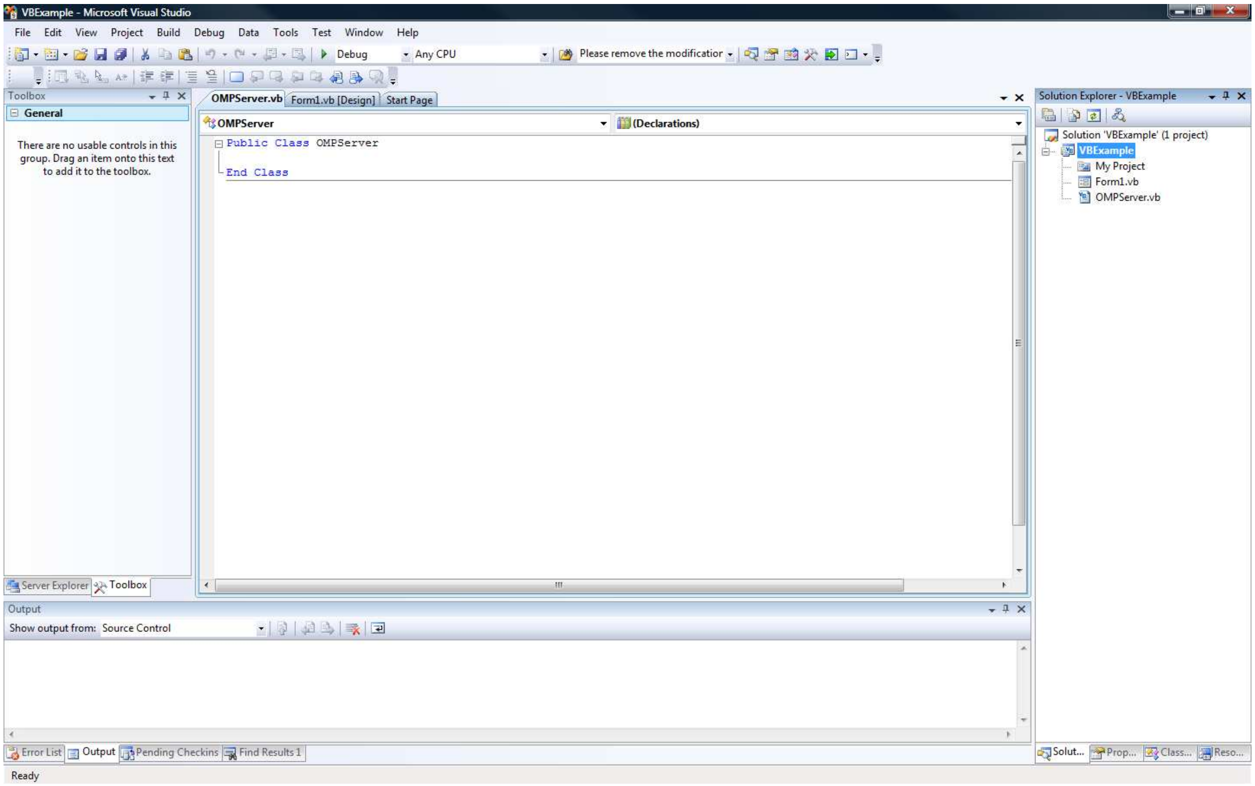
Run OMPServer:

1) Go to the directory containing OMPServer.exe (default is C:\Program Files\DNASoftware\OMP)
   a) To go to the default drive, type the following without quotation marks: “cd C:\”
   b) To go to the default directory, type the following without quotation marks: “cd Program Files\DNASoftware\OMP”
2) Type “OMPServer.exe /dlg”
3) Select the browse buttons to choose the appropriate input and output files or type the paths in the “Input File” and “Output File” fields
4) Press “Run”

1.2.3 Running OMP DE through the COM interface

Example projects of how to use the COM interface (both for C# and VB) are available at DNA Software and can be provided to users upon request.

Using Visual Studio .NET: C# or VB

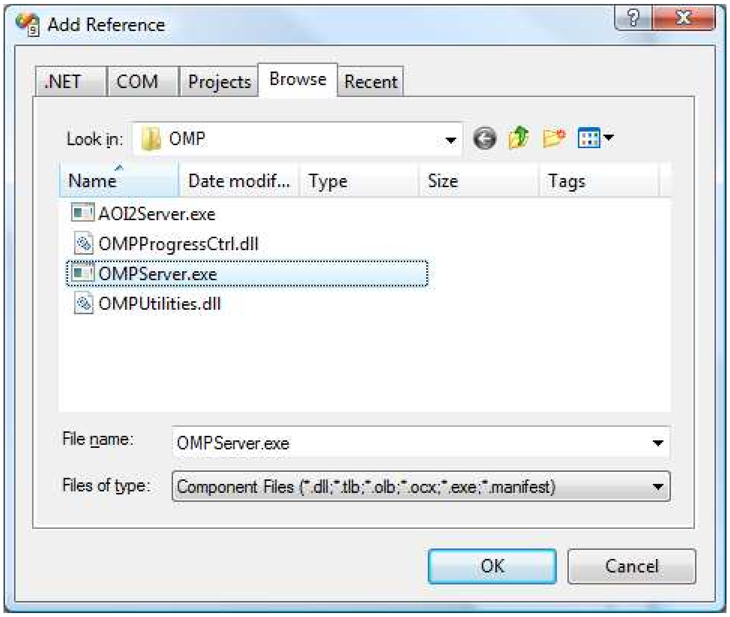


VBExample - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Any CPU
Please remove the modificatior
Toolbox
General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
OMPServer.vb
Form1.vb [Design]
Start Page
OMPServer
(Declarations)
Solution Explorer - VBExample
Solution 'VBExample' (1 project)
VBExample
My Project
Form1.vb
OMPServer.vb
Server Explorer
Toolbox
Output
Show output from: Source Control
Error List
Output
Pending Checkins
Find Results 1
Solut...
Prop...
Class...
Reso...
Ready

```
Public Class OMPServer

End Class
```

Right-click on the Project and select Add Reference. Select Browse and navigate to OMPServer.exe on the local hard-drive. (e.g. C:\Program Files\DNASoftware\OMP\OMPServer.exe)



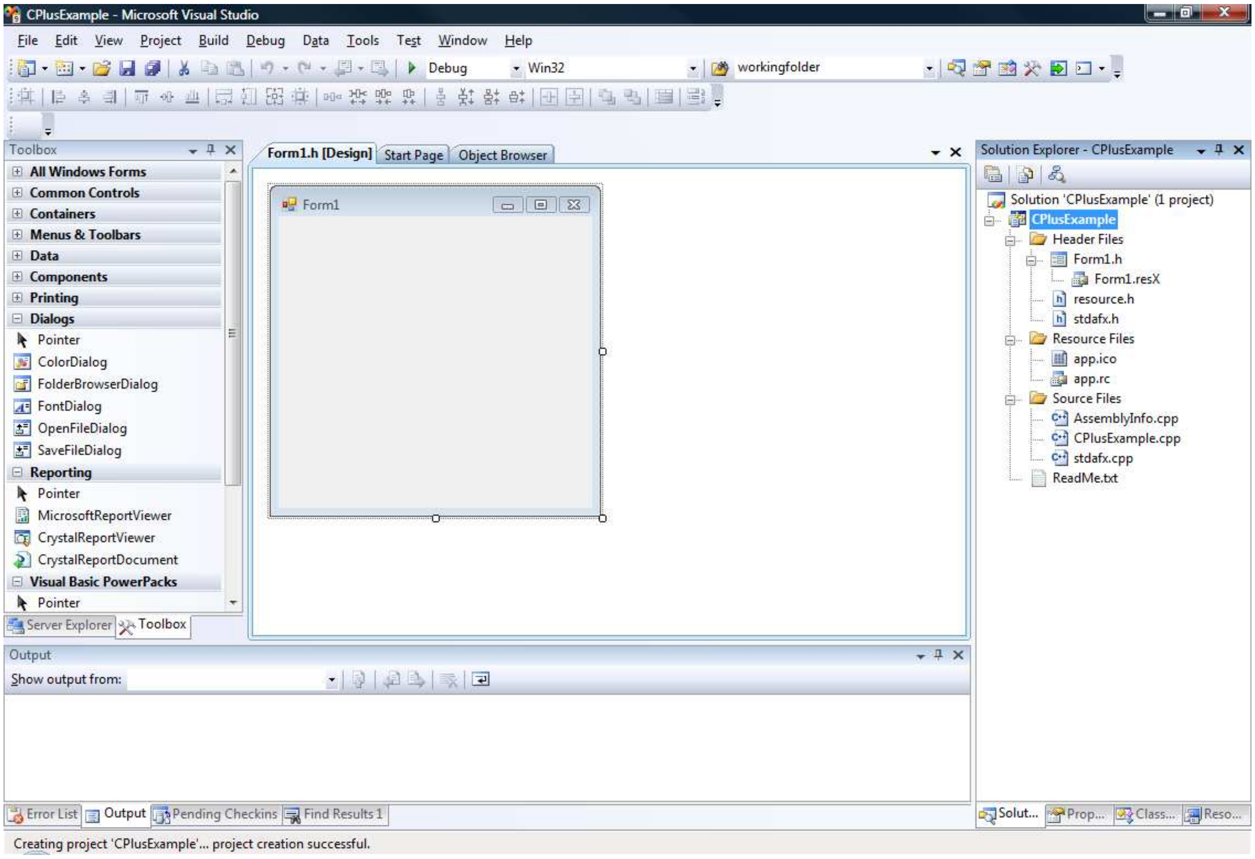
Look in: OMP

File name: OMPServer.exe

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

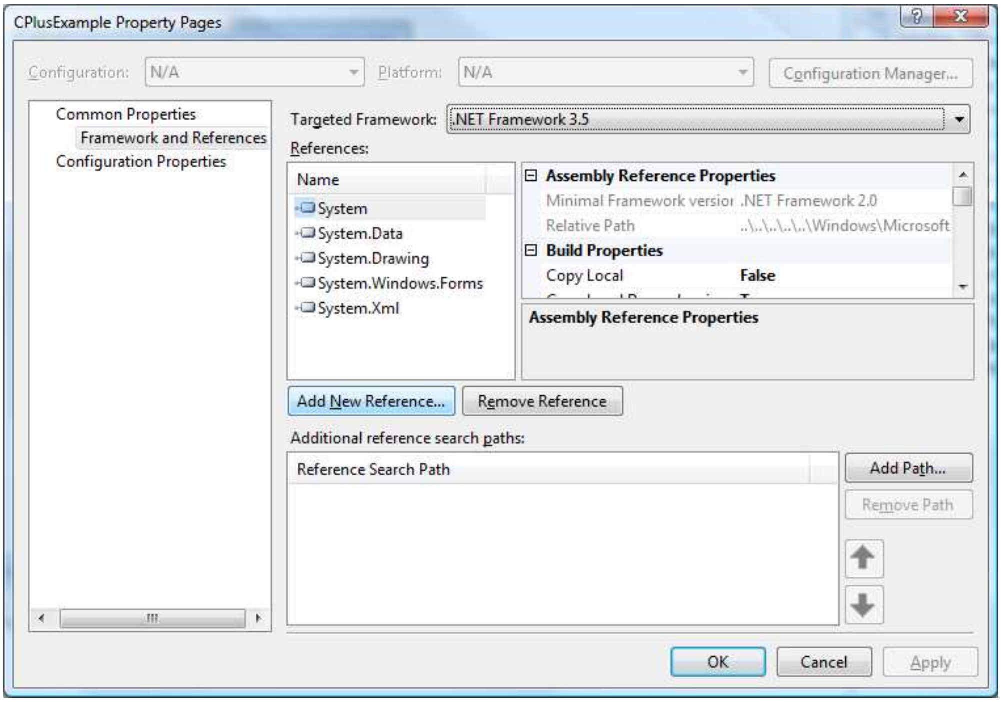
This creates the reference for OMPServer to be accessed via the COM Interface.
The following explains the functions and calls that the user can implement in their project: example files for Visual Basic and C# are available.

Using Visual Studio .NET: C++



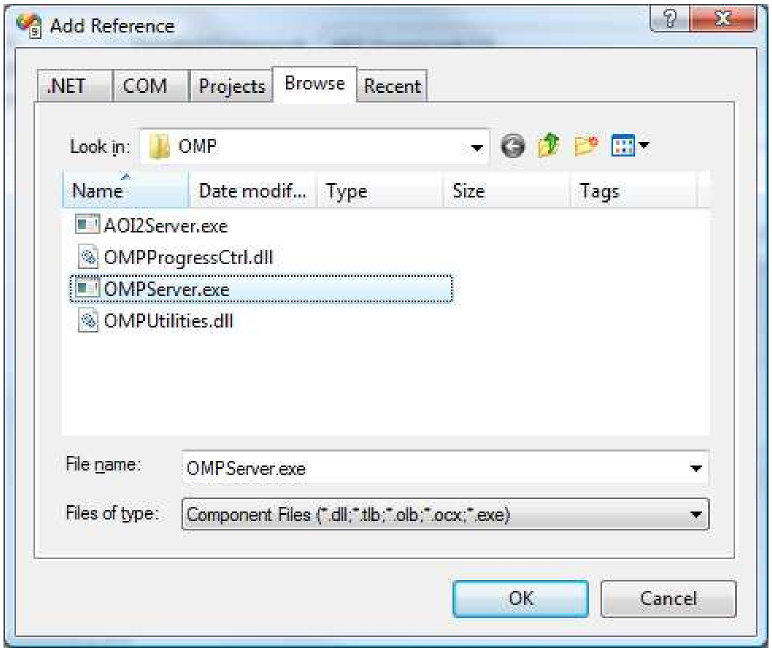
Creating project 'CPlusExample'... project creation successful.

Right Click Project Name and select References. Click the "Add New Reference" button as shown in the next screen shot.



Configuration: N/A
Platform: N/A
Targeted Framework: .NET Framework 3.5
References:
Minimal Framework version .NET Framework 2.0
Relative Path ..\..\..\..\..\Windows\Microsoft
Copy Local False

Select Browse and navigate to OMPServer.exe on the local hard-drive. (e.g. C:\Program Files\DNASoftware\OMP\OMPServer.exe)



Look in: OMP

AOI2Server.exe
OMPProgressCtrl.dll
OMPServer.exe
OMPUtilities.dll

File name: OMPServer.exe

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe)

Instantiation (VB):

```
        Public WithEvents objOMPServer as OMPSERVERLib.OMPInterface = New
OMPSERVERLib.OMPInterface()
```

Function	Description
ScheduleTask2	This function schedules a task to be run at a later time by OMPServer.exe. The function expects two parameters: The full path and name of the Input file, and the full path and name of the output file. A task identification number is returned by the function. OMPServer increments the task identification number by one from the last ID number in a controlled way.
ScheduleTask	This function schedules a task to be run at a later time by OMPServer.exe. The function expects three parameters: A task ID number to be specified by the user, the full path and name of the input file, and the full path and name of the output file. This function allows the user to control the task identification number.
RunTask	This function runs the scheduled task setup by the ScheduleTask2 or ScheduleTask functions. A task must be previously scheduled for RunTask to execute.
DequeueLocalTask	This function will remove a scheduled task in OMPServer by passing in the parameter Task ID into the function.
ResetRunTaskID	This function will update and change the identification Number of a task. This function has one parameter of TaskID which is the current task ID the user wishes to reset.
Run	This function runs an OMP simulation on one file. This function contains two parameters: The full path and file name of the input file, and the full path and name of the output file.
TerminateLocalTasks	This function will terminate all task that are queued to be run by OMPServer.exe
GetOMPLocation	This function will return the path of OMPServer as a string.
GetRegistrationInfo	This function returns information regarding the OMP.lic license associated with OMPServer.exe. The returned string contains information about:

	1) Date OMPServer.exe was first run. 2) Date OMPServer.exe was last run. 3) Date the OMP.lic was issued (effective date). 4) Days remaining in the license. 5) Application specificity. 6) Local Machine Identification. 7) OMPServer Version Number. 8) OMPServer Time Stamp.
GetScheduledTasks	This Function Returns a list of the scheduled tasks in the queue as a string.
NumOfLocalTasks	This Function returns the number of tasks set in the queue for OMPServer.exe.
ExitRunTask	This function will exit an experiment that is currently processing. The function expects a parameter, but the parameter is not functional and should have the value 0 passed into it.
IDRun	This function is used when not using any of the OMP Schedule Task functions to schedule jobs, but using your own system of scheduling and setting up jobs. This function expects three parameters: TaskID which is the ID of the task you are running (Integer), The full path and file name of the input file (String), and the full path and file name of the output file (String). When the job is completed the TaskID is returned to the function OMPServer_RunCompleted.
KillAll	This function kills all OMPServer process threads for local and distributed processing.
RunCompleted	This event function is an event that is fired from OMPServer upon a completion of a task that has finished running. The completed TaskID is passed into the function by OMPServer.exe. Add your code to this event function on what you want done once a job is completed.
StatusMessage	This event function is an even that is fired from OMPServer when a status message update is sent from the OMPServer.exe. That status message is passed into this function by OMPServer and the user may define what they wish to do with this status message.

***** Other functions that are available for user selection are not currently supported for COM users and should not be used in your program. If they are used they could result in undesired results or cause errors in your application.***

1.3 How to Run an Experiment (Linux)

OMP DE for Linux is composed mainly of two engines: OMP and AOI. OMP is used for simulation and analysis. All secondary structures, suboptimal structures, numerical analysis and main thermodynamics are computed by this engine. AOI is used for probe and/or primer design. Both OMP and AOI are run through the command line.

Open a terminal window and run OMP or AOI:

1) In the terminal window, navigate to the folder that contains the OMP and AOI executables (e.g. type in “cd /home/user/OMP_DE”)
2) Type “./OMP Input.oef Output.oof” where Input.oef is any input file and Output.oof is any output file to run OMP. Or type “./AOI Input.odf Output.osf” where Input.odf is any input file and Output.osf is any output file to run AOI.
If the input and output files are located in a directory other than the one containing OMP and AOI, their full paths need to be given in the run simulation / run design commands. Like for example: “./OMP /home/user/OMPData/input.oef /home/user/OMPData/output.oof” in case the input-file is saved in the /home/user/OMPData directory and the output-file needs to be saved in this same directory.

The rest of this OMP DE Manual is written referring to the Windows executables (OMPServer.exe and AOI2Server.exe); reader can replace these with OMP and AOI, respectively, if running the Linux OMP DE version (using the OMP and AOI executables). Keywords, input files and output files are equal across the Windows and Linux platforms.

1.4 Notes on OMPServer Input Files

Input text files are recognized by OMPServer by the “.oef” and “.sif” extensions (OMP Experiment File, Scorpion Input File). Comments are ignored by OMPServer and may be added in two ways: 1) notes within “[“ and “]” are ignored and 2) notes preceded by “//” are ignored.

Keywords are given in Boulder I/O format; this is defined by a Keyword, immediately followed by an “=” sign and then immediately followed by a value.

Input sections are subdivided into five sections:

- The **project information** section provides overall information and conditions.
- The **solution conditions** section lists the conditions of the solution.
- The **defaults** section establishes default values but these can almost always be individually overridden in the sequences section.
- The **sequences** section provides information about all sequences, concentrations and sequence identities.
- The **NetTm** section provides an area in which users may customize melting temperature queries.

1.5 Notes on OMPServer Output Files

OMPServer recognizes output text files by the “.oof”, “.nal”, “.tbs”, “.ta” and “.sof” extensions (OMP Output File, Numerical Analysis, Traceback Structure, Target Accessibility/Complexity, and Scorpion Output File). Keywords are given in Boulder I/O format; this is defined by a Keyword, immediately followed by an “=” sign and then immediately followed by a value.

Output files for a normal OMP experiment begin with project information and then are structured from simplest species (monomers) to more complex species (e.g. n-plexes). In general, each species is subdivided into optimal and suboptimal structures. Each structure is then further divided into thermodynamic characteristics and numerical analysis at the assay temperature.

1.6 Notes on AOI2Server Input Files

Input text files are recognized by AOI2Server by the “.odf” extension (OMP Design File). Comments are ignored by AOI2Server and may be added in two ways. 1) notes within “[“ and “]” are ignored and 2) notes preceded by “//” are ignored.

Keywords are given in Boulder I/O format; this is defined by a Keyword, immediately followed by an “=” sign and then immediately followed by a value.

Input sections are subdivided into four sections:

- The **project information** section provides overall information.
- The **solution conditions** section lists the conditions of the solution.
- The **sequences** section provides information about all sequences, concentrations and sequence identities.
- The **designs** section provides an area in which users define parameters for the design of single or multiple oligos.

1.7 Notes on AOI2Server Output Files

AOI2Server recognizes output text files by the “.osf” extension (OMP Solution File). Keywords are given in Boulder I/O format; this is defined by a Keyword, immediately followed by an “=” sign and then immediately followed by a value.

AOI2Server design output files contain a relatively straightforward architecture. The first half of the output file contains all of the parameters from the design input file (ODF) used to determine the probes and/or primers. The keywords and explanations for each can be found in chapter 3.2 AOI2Server Design. Directly following the parameters is a section that summarizes the design experiment. Lastly, the solutions to the design are listed. This documentation only shows the output for a single solution; however, most outputs will likely contain more than one solution. If a multiplex design is performed, two oligos with different types (e.g. primer pair and probe) may have the same solution number.

1.8 Working with files

The best way to work with OMPServer or AOI2Server files is to use examples from the website or to use samples from Visual OMP. All OMPServer input, OMPServer output, secondary structure, and numerical analysis, AOI2Server input, and AOI2Server output files are available using the Visual OMP interface. To view the source files for any action (such as input), from within the Visual OMP package, right click on any input or output grids and select the “View Source” option. Similarly, right click on any numerical analysis or secondary structure to view those particular source files.

Using the Visual OMP program to understand how OMP DE works and to make template files is the easiest way to get started with OMP DE!

Chapter 2. Input (OEF, ODF) File Keywords

2.1 General OMPServer Input Files

A normal OMPServer experiment calculates the thermodynamics of a single or multi-state equilibrium of various types of oligonucleotides. In the simplest case, a single DNA sequence may be run at a known NaCl and $MgCl_2$ concentration and at a known temperature. However, this is usually too basic for the average user. Most normal OMP experiments involve at least 2 oligonucleotides and measures a) the behavior of the oligos with each other, b) the behavior of the oligos with themselves and c) their behavior at differing temperatures.

A simple OMPServer file may be broken down into 4 parts: A "project information" section, a "solution" section, a "defaults" section and a "sequences" section. The first section contains information about the project: the name of the project, the version, the user as well as technical details about the experiment to be run. This can include information about whether to output a numerical analysis, the output directory for files, solution conditions, etc.

The solution section contains the information on the solution conditions including temperature, salt concentrations, and buffer concentrations.

The defaults section defines global constraints such as COMPLEMENT_DEFAULT=true, REVERSE_SEQUENCE_DEFAULT=true. For example, if these 2 keywords are defined, then all oligos will be defaulted to be the reverse complement of whatever is written.

Lastly, the sequence section holds information about the oligos themselves. Each sequence has an identifier, a literal sequence and an appropriate concentration. These are the minimal necessities.

2.1.1 Project Information Section

[project information]

Describes general information about the OMP simulation including identification and output conditions of the simulation.

DESCRIPTION=This experiment simulates the hybridization between one target and one probe taking into account the competing monomer and homodimer structures.

A description of the OMP experiment can be given by using this keyword.

NAME=SimulationOne

The file can be given a name (1000 character limit).

VERSION=1/29/2006 3:34:33 PM

The version of the project that is using OMP.

USER_NAME=JohnnyBGood

The user of the file (1000 character limit) can be specified.

GENERATE_LOG=true

GENERATE_LOG accepts Boolean values. If *true*, then a log of OMP's activities is output to the install directory (the directory where OMPServer.exe is located). It is recommended that beginners always output a log; the name of this log is AppLog1.

OUTPUT_DIRECTORY=C:\OMPData

All output files can be found in this directory. If the directory is not specified or if the directory does not exist, output files can be found in the same directory as the application file (OMPServer.exe)

ECHO_INPUT=true

If this keyword is set to true, then information from the input file will be copied onto the output file. Some users may prefer to do this so that input and output files are “married” for further use. It is not recommended when OMP is automated. Default setting is false.

DEFAULT_GENERATE_NETTM=true

Sets the default as to also calculate (if set to *true*) the Net Tm and Net dG, besides the 2-state dG and Tm.

TARGET_THERMO_ACCESS_WINDOW=20

For sequences of SEQUENCE_FUNCTION=TARGET, OMPServer will calculate the energy needed to unfold target regions. The size of these regions (in nucleotides) is set by this TARGET_THERMO_ACCESS_WINDOW keyword. The output of these calculations can be found in the .ta output file.

TARGET_COMPLEXITY_WINDOW=20

For sequences of SEQUENCE_FUNCTION=TARGET, OMPServer will calculate the variety in bases for individual target regions. The size of these regions (in nucleotides) is set by this TARGET_COMPLEXITY_WINDOW keyword. The output of these calculations can be found in the .ta output file.

GENERATE_NUMANALY=true

This switch controls the numerical analysis output for OMP. If *false*, the numerical analysis is not written to an output file (.nal, NUMANALY_OUTFILE) described below. If *true*, the numerical analysis is output from 100˚ Celsius to 10˚ Celsius in case the minimum and maximum temperature are not specified by the NUMANALY_MIN_TEMPERATURE and NUMANALY_MAX_TEMPERATURE keywords (see below).

NUMANALY_OUTFILE=numanaly.txt

If this keyword is not specified and GENERATE_NUMANALY=true, the numerical data will be written to a .nal file that has the same name as the experiment itself (i.e. test.oef will generate a test.nal file when run with OMPServer). Using this keyword, the name of the text file to which all numerical analysis data is outputted can be specified. The data is delimited with tabs. The location of the file is the same as specified by the OUTPUT_DIRECTORY described above.

NUMANALY_MIN_TEMPERATURE=10

The minimum temperature used in the numerical analysis. If the NUMANALY_MIN_TEMPERATURE=10, then the numerical analysis will be run from 100ºC to 10ºC.

NUMANALY_MAX_TEMPERATURE=95

The maximum temperature used in the numerical analysis. If the NUMANALY_MAX_TEMPERATURE=95, then the numerical analysis will be performed from 95ºC to 0ºC.

CONVERGENCE_CYCLES=100000

The maximum number of cycles per degree that a numerical analysis is allowed to run before breaking out unsuccessfully. It is possible to avoid long computational times by allowing only a certain number of CONVERGENCE_CYCLES to occur per degree. Therefore, if it takes more than 10000 cycles to determine the concentration of species, then the simulation will prematurely end.

CONVERGENCE_TIMEOUT=100

The maximum amount of time per degree that a numerical analysis is allowed to run before breaking out unsuccessfully. It is possible to avoid long computational times by allowing only a certain amount of time, CONVERGENCE_TIMEOUT, to elapse per degree. Therefore, in the example, if it takes more than 100 seconds to determine the concentration of species, then the simulation will prematurely end.

DELTA_CP=0.05

Gives users the possibility to apply a change in heat capacity correction to account for single strand stacking of oligonucleotides at low temperatures. A correction of 0.05 is recommended; the default is set to 0.

CALCULATE_EXTENSIBILITY=true

A species is considered extensible by OMPServer if it is likely to be extended by DNA polymerase. If this keyword is set to *true*, structures that are extensible will be recognizable in the OMP Output file.

Through the keywords EXTENSION_OVERHANG, EXTENSION_WINDOW, and EXTENSION_MIN_PAIRS one can set the conditions as to what kind of structures should and what kind of structures should not be labeled “extensible”.

EXTENSION_OVERHANG=1

In the following example, a monomer is shown to have an extension overhang of 1 (in red). Some polymerases with exonuclease activity do not require a blunt ended 3’ terminus, thus making the identification of species (monomer, homodimer, heterodimer etc.) with a certain overhang significant. When the keyword EXTENSION_OVERHANG is used, a species with an overhang on the 3’ end between 0 and the value of the keyword will be marked as “extensible” in the OMP output file. Setting EXTENSION_OVERHANG to 0 will only flag 3’ blunt ends as being extensible.

If this keyword isn’t specified and CALCULATE_EXTENSIBILITY=true, the default value of 1 will be used for this keyword.

```
 ____________ACGTA   3’
/             | | | |
\____________TGCACCGATGGAGTA   5’
```

EXTENSION_WINDOW=3

A minimum template length for a species to be considered “extensible” by polymerase can be specified with this keyword. In this example of value 3, 3 or more nucleotides need to be available on the strand to which the 3’ extensible end is hybridized. These nucleotides then function as the extension template.

If this keyword isn’t specified and CALCULATE_EXTENSIBILITY=true, the default value of 3 will be used for this keyword.

EXTENSION_MIN_PAIRS=4

A minimum number of basepairs need to be formed between the oligo’s 3’ end and the template strand it hybridizes to. Matching basepairs and single internal mismatches are counted towards this keyword.

If this keyword isn’t specified and CALCULATE_EXTENSIBILITY=true, the default value of 4 will be used for this keyword.

MULTIPLEX_ENABLE=TRUE

If MULTIPLEX_ENABLE=TRUE, then OMPServer will perform a multiplex against all eligible heterodimers. A heterodimer is considered eligible if it has a certain energy which is defined by MULTIPLEX_ENERGY_WINDOW or MULTIPLEX_ENERGY_THRESHOLD. For all eligible heterodimers, the multiplex will determine trimers and tetramers between targets and probes/primers. Currently, the trimers and tetramers must consist of a single target and multiple probes and/or primers (defined by the SEQUENCE_FUNCTION keyword, see below).

MULTIPLEX_ENERGY_WINDOW=80

Multiplex interactions are taken from eligible heterodimers. For a heterodimer to be considered eligible, it must have a certain energy which is defined by MULTIPLEX_ENERGY_WINDOW or MULTIPLEX_ENERGYTHRESHOLD. After a simulation is performed, the heterodimer with most negative dG (kcal/mol) is considered along with all other heterodimers that contain the same target sequence that are within the MULTIPLEX_ENERGY_WINDOW. Therefore, in a simulation, if a heterodimer with the most negative dG had a dG of -10 kcal/mole and the MULTIPLEX_ENERGY_WINDOW=80, then all other heterodimers within 80 percent of the top dG would be considered. (Heterodimers between -8 and -10 would be considered, but heterodimers more positive than -8 would not be considered.)

MULTIPLEX_ENERGY_THRESHOLD=5

Multiplex interactions are taken from eligible heterodimers. For a heterodimer to be considered eligible, it must have a certain energy which is defined by MULTIPLEX_ENERGY_WINDOW or MULTIPLEX_ENERGYTHRESHOLD. After a simulation is performed, the heterodimer with most negative dG (kcal/mol) is considered along with all other heterodimers that contain the same target sequence that are within the MULTIPLEX_ENERGY_THRESHOLD. Therefore, in a simulation, if a heterodimer with the most negative dG had a dG of -10 kcal/mole and the MULTIPLEX_ENERGY_THRESHOLD=5, then all other heterodimers within 5 kcal/mole would be considered. (Heterodimers between -5 and -10 would be considered, but heterodimers more positive than -5 would not be considered.)

2.1.2 Solution Section

[solution]

In this section the conditions of the solution can be set, like temperature, salt and buffer concentrations.

ASSAY_TEMPERATURE=37

Assay temperature is measured in degrees Celsius. The valid assay temperature range is from 0ºC to 100ºC. However, the most accurate values will be localized around an assay temperature of 50ºC. Simulations with assay temperatures between 20-80ºC are still reliable however, tend to drop off in reliability outside of these ranges.

MAGNESIUM_CONCENTRATION=0.5

Magnesium concentration is measured in M. Mg concentration ranges from 0 to 1M are reliable, and OMP can take higher values. However, reliability decreases with the increase of this salt.

SODIUM_CONCENTRATION=1

Sodium concentration is measured in M. Sodium concentration ranges from 0.04 to 1M are reliable, and OMP can take higher values. However, reliability decreases with the increase of this salt.

GLYCEROL_CONCENTRATION=2

Concentration of the buffer glycerol, measured in M.

GLYCEROL_CONC_UNITS=PERCENT

In Visual OMP glycerol units can be displayed in M (default if this keyword isn't specified) or in percentage. If this keyword is set to the value *percent*, then from the GLYCEROL_CONCENTRATION (in M) keyword, it will be calculated what the equivalent in percent is and this value will be displayed in the User Interface of Visual OMP.

DMSO_CONCENTRATION=3

Concentration of the buffer DMSO (dimethyl sulfoxide), measured in M.

DMSO_CONC_UNITS=PERCENT

In Visual OMP DMSO units can be displayed in M (default if this keyword isn't specified) or in percentage. If this keyword is set to the value *percent*, then from the DMSO_CONCENTRATION (in M) keyword, it will be calculated what the equivalent in percent is and this value will be displayed in the User Interface of Visual OMP.

FORMAMIDE_CONCENTRATION=4

Concentration of the buffer formamide, measured in M.

FORMAMIDE_CONC_UNITS=percent

In Visual OMP formamide units can be displayed in M (default if this keyword isn't specified) or in percentage. If this keyword is set with the value *percent*, then from the FORMAMIDE_CONCENTRATION (in M) keyword, it will be calculated what the equivalent in percent is and this value will be displayed in the User Interface of Visual OMP.

TMAC_CONCENTRATION=1

Concentration of the salt TMAC (Tetramethyl ammonium chloride), measured in M.

BETAINE_CONCENTRATION=0.5

Concentration of the zwitter-ion Betaine, measured in M.

POLYMER_SALT=true

Polymer duplexes (greater than 16 base pairs) have a different salt dependence than oligomers due to end effects and counterion condensation effects. If this keyword is set to TRUE (which is recommended best practice), then the software applies the polymer salt correction to all duplexes that are longer than 16 basepairs, while shorter duplexes and unimolecular folds are unaffected.

MICROCHIP_CORRECTION=SantaLucia corrections

A linear correction to solution thermodynamics to allow good agreement between solution predictions and experiments in microarrays that use gel pads can be specified. The value for this keyword can be a string describing the correction. The actual correction is defined by setting the next 4 keywords:
SURFACE_SLOPE_DELG
SURFACE_INTERCEPT_DELG
SURFACE_SLOPE_DELH
SURFACE_INTERCEPT_DELH
Following chart gives the values of these keywords for the Fotin et al. Corrections (see the original article: http://www.dnasoftware.com/Science/Publications/pdf/microchips.pdf) and the SantaLucia Corrections (same as Fotin, but correcting a type in the original paper). If one chooses not to use a microarray correction, these keywords can be omitted or the values for No Corrections can be used. You can also use your own values for the correction if you know the values for your specific microarray.

	Fotin et al. Corrections	SantaLucia Corrections	No Corrections
SURFACE_SLOPE_DELG	1	0.85	1
SURFACE_INTERCEPT_DELG	3.2	2.33	0
SURFACE_SLOPE_DELH	1	1	1
SURFACE_INTERCEPT_DELH	24.0	24.0	0

SURFACE_SLOPE_DELG=.85

The slope of the delta G microarray correction at 37 degrees and 1M NaCl.

SURFACE_INTERCEPT_DELG=2.33

The intercept of the delta G microarray correction at 37 degrees and 1M NaCl.

SURFACE_SLOPE_DELH=1

The slope of the delta H microarray correction at 37 degrees and 1M NaCl.

SURFACE_INTERCEPT_DELH=24.0

The intercept of the delta H microarray correction at 37 degrees and 1M NaCl.

2.1.3 Defaults Section

[defaults]

The defaults section allows the user to input global settings such as complement_default. If this is declared in the default section as false, then this parameter applies to all strands. Conversely, each of the default parameters can be deleted from the “defaults” section and then the sequence specific keyword can be individually placed into the specific sequence section. If the default parameter is left in the defaults section, it can still be overridden by specifying the individual parameter in the specific sequence section.

FIXED_DEFAULT=false

A sequence can either be fixed (as on a microarray surface) or free (as in solution). If fixed_default is false, then the sequence is considered free. These are default values and the fixed condition may be changed for any sequence.

STRAND_DEFAULT=single

Sequences may be single or double stranded. These are default settings and each sequence may be identified as single or double stranded.

COMPLEMENT_DEFAULT=false

If this is set to true, then the complement (A->T, C->G, G->C, T->A) of the sequences will be run through OMP. This parameter can be changed for each individual sequence. The default is false.

REVERSE_SEQUENCE_DEFAULT=false

If this is set to true, then the reverse (backward) of the sequences will be run through OMP. This parameter can be changed for each individual sequence. The default is false.

DEFAULT_GENERATESTRUCTURE_MONOMER=true

The generate structure for monomers input generates the Traceback data for optimal and suboptimal structures for all monomers. These files (.tbs) are outputted to the output directory described above. Details of structures presently exist as text descriptions of base pairings in these .tbs-files and can be visualized through Visual OMP. Default is false.

DEFAULT_GENERATESTRUCTURE_HOMODIMER=true

Similar to DEFAULT_GENERATESTRUCTURE_MONOMER, but applied to homodimer (=self-dimer) species).

DEFAULT_GENERATESTRUCTURE_HETERODIMER=true

Similar to DEFAULT_GENERATESTRUCTURE_MONOMER, but applied to heterodimer (=duplex) species.

DEFAULT_GENERATE_NETTM=TRUE

If set to true, a NETTM with a NETTM_THRESHOLD=50 will be generated for all species in the OMP output. The default value is “FALSE”.

2.1.4 Sequences Section

[sequences]

This section lists all of the sequences to be run in one OMP simulation.

OPTIMAL_ENERGY_THRESHOLD_MONOMER=1

Users may filter insignificant monomer species in the output files by using this keyword. In this example, monomers which have a deltaG of greater than 1 kcal/mole will be excluded from the output.

OPTIMAL_ENERGY_THRESHOLD_HOMODIMER=2

Users may filter insignificant homodimer species in the output files by using this keyword. In this example, homodimers which have a deltaG of greater than 2 kcal/mole will be excluded from the output.

OPTIMAL_ENERGY_THRESHOLD_HETERODIMER=2.5

Users may filter insignificant heterodimer species in the output files by using this keyword. In this example, heterodimers which have a deltaG of greater than 2.5 kcal/mole will be excluded from the output.

ILMAX_MONOMER=10

Maximum Internal Loop Length for monomer structures. The maximum internal loop and bulge settings are intended to control how much OMP investigates internal secondary structures. The smaller the values of the settings, the less exhaustive OMP will treat the simulation. Manipulation of these settings is only recommended in cases where targets are of extremely long lengths.

ILMAX_HOMODIMER=10

Maximum Internal Loop Length for homodimer structures.

ILMAX_HETERODIMER=10

Maximum Internal Loop Length for heterodimer structures.

BULGEMAX_MONOMER=7

Maximum Bulge Loop Length for monomer structures. The maximum bulge and internal loop settings are intended to control how much OMP investigates internal secondary structures. The smaller the values of the settings, the less exhaustive OMP will treat the simulation. Manipulation of these settings is only recommended in cases where targets are of extremely long lengths.

BULGEMAX_HOMODIMER=7

Maximum Bulge Loop Length for homodimer structures.

BULGEMAX_HETERODIMER=7

Maximum Bulge Loop Length for heterodimer structures.

SUBOPTIMAL_ENABLE=true

Besides the optimal structure of a species (= the most energetically favorable configuration with the lowest ∆G) OMP is able to also calculate structures that can be present at a lower percentage and of which the thermodynamics are close to the optimal structure, called suboptimal structures.

The following keywords define the energy range between optimal and suboptimal structures and how many suboptimal structures OMP will output.

MAX_STRUCTURES_MONOMER=15

The maximum number of monomer suboptimal structures allowed. MAX_STRUCTURES_MONOMER is a hard limit and in the example, OMP will not return more than 15 monomer suboptimal structures.

MAX_STRUCTURES_HOMODIMER=10

The maximum number of homodimer suboptimal structures allowed. MAX_STRUCTURES_HOMODIMER is a hard limit and in the example, OMP will not return more than 10 homodimer suboptimal structures.

MAX_STRUCTURES_HETERODIMER=12

The maximum number of heterodimer suboptimal structures allowed. MAX_STRUCTURES_HETERODIMER is a hard limit and in the example, OMP will not return more than 12 heterodimer suboptimal structures.

SUBOPTIMALWINDOW_MONOMER=50

There are three ways to describe how suboptimal structures should be reported (SUBOPTIMALWINDOW, SUBOPTIMALDISTANCE, and SUBOPTIMALENERGY); it is recommended that only advanced users use more than one descriptor at a time. The SUBOPTIMALWINDOW descriptions calculate the amount of structures by percentage of the optimal structure's ∆G that OMP should report to the user using traceback. In the example here, the suboptimal window for all monomers is set to 50%: if the Optimal (most likely structure) ∆G for the sequence is -5 kcal, then OMP will return all structures with a ∆G between -2.5 and –5 kcal.

SUBOPTIMALWINDOW_HOMODIMER=50

Similar to SUBOPTIMALWINDOW_MONOMER.

SUBOPTIMALWINDOW_HETERODIMER=100

Similar to SUBOPTIMALWINDOW_MONOMER. If no heterodimer exists, then the program will exit early.

SUBOPTIMALDISTANCE_MONOMER=3

There are three ways to describe how suboptimal structures should be reported (SUBOPTIMALWINDOW, SUBOPTIMALDISTANCE, and SUBOPTIMALENERGY); it is recommended that only advanced users use more than one descriptor at once. The distance function is a measure of similarity. It can be defined as the minimum number of insertions, deletions and substitutions required to convert structure A to structure B. A value input for this keyword enables the user to define the minimum difference between an optimal structure and the first suboptimal structure. In the example, the first MONOMER suboptimal structure will have a minimum distance function of 3 between it and the MONOMER optimal. Similarly, the second MONOMER suboptimal will have a minimum distance function of 3 between it and the first MONODIMER suboptimal, and so on.

SUBOPTIMALDISTANCE_HOMODIMER=3

Similar to SUBOPTIMALDISTANCE_MONOMER. If no homodimer exists, then the program will exit early.

SUBOPTIMALDISTANCE_HETERODIMER=3

Similar to SUBOPTIMALDISTANCE_MONOMER. If no heterodimer exists, then the program will exit early.

SUBOPTIMALENERGY_WINDOW_MONOMER=1

There are three ways to describe how suboptimal structures should be reported (SUBOPTIMALWINDOW, SUBOPTIMALDISTANCE, and SUBOPTIMALENERGY); it is recommended that only advanced users use more than one descriptor at once. SUBOPTIMALENERGY_WINDOW is the maximum energy in kcal/mole that the ΔG of a suboptimal structure can be away from the optimal structure at the ASSAY_TEMPERATURE. Therefore in the example, if an optimal structure for a target is -5.67 kcal, then OMP will return all suboptimal structures between -5.67 and -4.67 kcal.

SUBOPTIMALENERGY_WINDOW_HOMODIMER=1

Similar to SUBOPTIMALENERGY_WINDOW_MONOMER.

SUBOPTIMALENERGY_WINDOW_HETERODIMER=1

Similar to SUBOPTIMALENERGY_WINDOW_MONOMER.

REPORTING_PERCENT_BOUND_MONOMER=5

When performing larger OMP experiments with significant reactions/species, such as crossdimerization or microarray simulation, it is possible that reporting every species may be extraneous. If implemented, the keyword: REPORTING_PERCENT_BOUND_MONOMER= 5 will tell OMP to only report (in the .OOF file) only those monomers which are more than 5% bound (or folded, in the case of Monomer). Homodimers and heterodimers are not affected.

REPORTING_PERCENT_BOUND_HOMODIMER=7

When performing larger OMP experiments with significant reactions/species, such as crossdimerization or microarray simulation, it is possible that reporting every species may be extraneous. If implemented, the keyword: REPORTING_PERCENT_BOUND_HOMODIMER= 7 will tell OMP to only report (in the .OOF file) only those homodimers which are more than 7% bound. Monomers and heterodimers are not affected.

REPORTING_PERCENT_BOUND_HETERODIMER=9

When performing larger OMP experiments with significant reactions/species, such as crossdimerization or microarray simulation, it is possible that reporting every species may be extraneous. If implemented, the keyword: REPORTING_PERCENT_BOUND_HETERODIMER= 9 will tell OMP to only report (in the .OOF file) only those heterodimers which are more than 9% bound. Monomers and homodimers are not affected.

SEQUENCE_NAME=Target-1

This is the name of the sequence.

SEQUENCE=ACAACAGAAGCTGACCTCTTTGATCTCTTGCGCAG

This is the actual base order of the sequence. OMP does not currently support IUPAC ambiguity codes specified in the SEQUENCE keyword itself, but variations can be declared by the VARIATION_NAME, VARIATION_POSITION, VARIATION_TYPE, VARIATION_SEQUENCE, and VARIATION_LENGTH keywords described below.

If the Modifieds module is licensed, following modified nucleotides can be entered in sequences:

- deoxyUracil in DNA type sequences: enter “U” at desired position
- deoxyInosine in DNA type sequences: enter “I” at desired position
- 5-Mehtyl-cytosine in DNA type sequences: enter “{D_5mC}” at desired position
- Iso-Cytosine / Iso-Guanosine in DNA type sequences: enter “{DisoC}” / “{DisoG}” at desired position
- LNA in DNA type sequences: enter “{LNA_A}”, “{LNA_C}”, “{LNA_G}”, or {LNA_T}” at desired position
- 5-Methyl-Uridine in RNA type sequences: enter “T” at the desired position
- LNA in RNA type sequences: enter “{LNA_A}”, “{LNA_C}”, “{LNA_G}”, or {LNA_U}” at desired position

SEQUENCE_RANGE=300-500

The SEQUENCE_RANGE represents a subset of the total sequence. For example, it is possible to “load” a 1000mer sequence and then only consider a smaller fragment. In the above example, the sub-fragment would start at nucleotide 300 from the 5’ end and end at nucleotide 500.

CONCENTRATION=0.000000025

There is no maximum concentration for any sequence at this time. However, the concentration can not be 0, or less than 0. If a sequence is given a name, then OMP will assume that a species exists, and therefore a concentration of 0 which implies that the species does not exist will cause OMP to fail.

SEQUENCE_TYPE=DNA

The nature of the sequence. The SEQUENCE_TYPE may be DNA or RNA, and if the Modifieds module is licensed, PNA (Peptide Nucleic Acid), Mor (Morpholino), OM2 (2’-O-Methyl), and PSD (phosphorothioate) are options too. The default value is DNA.

OLIGO_ENABLED=TRUE

All sequences may be enabled or disabled. If a sequence has OLIGO_ENABLED=TRUE, then the sequence is considered in the simulation. If OLIGO_ENABLED=FALSE, then the sequence is not considered in the simulation.

STRAND=SINGLE

Whether the sequence in single (value is SINGLE) or double (value is DOUBLE) stranded. If a sequence is double stranded, the antisense strand is automatically generated and used during simulations. The interaction between sense and antisense strand is excluded from simulations with OMPServer.

FIXED=TRUE

A sequence may be declared as a fixed probe by setting this keyword to TRUE. Those sequences are considered to be “fixed” to a surface and will not interact with any other sequences that also have this keyword set to TRUE.

SEQUENCE_FUNCTION=TARGET

A sequence can be one of the following: TARGET, PROBE, PRIMER or BEACON.

EXCLUDE_SPECIES_TYPE=probe6,probe

This keyword is sequence specific and describes types of species with which a specific sequence may not interact. Therefore, in the example above, probe6 is specifically excluded from interacting with any other probe. The usage of this keyword is a specific sequence followed by a general type of species (either “target” or “probe”). This keyword may be particularly useful for microarrays.

EXCLUDE_SPECIES=target1,target1

This keyword is sequence specific and can describe certain species that will not be considered in the simulation. In the example above, the self-dimer (or homodimer) structure of target1 will be excluded from the simulation. If the keyword is set to “target1” only, this means that the monomer structure of target1 will be excluded from the simulation.

TAIL_FOLDING=TRUE

When set to TRUE this keyword indicates that for the monomer, homodimer and heterodimer species this sequence will be folded and the thermodynamics of the folded structure will be calculated. To save computational time or to only obtain the thermodynamics of the binding of a probe/primer to this sequence, the keyword can be set to FALSE. It will then be treated as a straight sequence without secondary structure.

INTENDED_TARGET=TargetA

If a probe or a primer has been designed against a certain target, it is possible to identify the target under the description of the probe or primer. In the output, it will be possible to identify the heterodimer of a correctly bound targetA by this probe/primer. The INTENDED_TARGET keyword is meant to organize the output into desirable/undesirable interactions.

INTENDED_TARGET_POSITION=14

If a probe or a primer has been designed against a certain target at a certain position from the 5’ end of the target, it is possible to identify the target position under the description of the probe or primer. In tandem with the INTENDED_TARGET, the INTENDED_TARGET_POSITION is meant to correctly identify heterodimers with correctly bound target and probe/primer. The INTENDED_TARGET and INTENDED_TARGET_POSITION keyword are meant to organize the output into desirable/undesirable interactions.

VARIATION_NAME=FIRSTVARIATION

The user defined name of the variation.

VARIATION_POSITION=15

A variation is defined as an insertion, deletion, or substitution mutation. The VARIATION_POSITION indicates the first nucleotide of the specific variation.

VARIATION_TYPE=Substitution

VARIATION_TYPE has three legal values: Substitution, Deletion or Insertion.

VARIATION_SEQUENCE=M

For substitution variations, the VARIATION_SEQUENCE may be a IUPAC ambiguity code or an actual sequence (eg. TGTGA). For deletion variations, the VARIATION_SEQUENCE value must be left blank. For insertion variations, the base or sequence of the insert must be declared (e.g. AGGT).

VARIATION_LENGTH=1

For Substitutions of a specific sequence (e.g. ACGT) the variation length needs to be set to the number of substituted nucleotides (4 in this example). For Deletions, variation length needs to be set to the number of nucleotides that are missing in the mutant allele with respect to the wild type allele.

SEQUENCE_MODIFICATION_NAME=myMOD

The user defined name of the modification, modifications for this and the following 2 keywords are understood to refer to fluorophore and quencher labels.

SEQUENCE_MODIFICATION_TYPE=Alexa350

The desired quencher or fluor. The options are listed in the fluorquencher.txt file that can be found in the C:\Program Files\DNASoftware\OMP folder (for Visual OMP) or the folder where the executables (OMPServer/OMP) are installed for OMP DE on Windows/Linux.

SEQUENCE_MODIFICATION_POSITION=4

The modification position defines the nucleotide that contains the specific modification.

LINKER_TYPE=glygly

If there is a linker present in the sequence it is defined here. Currently, OMP only supports one linker (gly-gly)

LINKER_POSITION=5

This is the position of the linker as read from the 5' of the sequence.

GENERATESTRUCTURE_MONOMER=true

This keyword, when placed under a specific sequence, will generate a traceback structure (.tbs) for the monomer of the sequence.

GENERATESTRUCTURE_HOMODIMER

This keyword, when placed under a specific sequence, will generate a traceback structure (.tbs) for the homodimer of the sequence.

GENERATESTRUCTURE_HETERODIMER

This keyword, when placed under a specific sequence, will generate a traceback structure (.tbs) for the heterodimer of the sequence.

2.1.5 NetTm Section

[NETTM]

Net Tm is declared in its own block, [NETTM]. NetTM is considered to be the temperature at which a known percent (usually 50%) of a limiting sequence is bound in the total complex. For example: A NetTM for the heterodimer, Target-1, Probe-2 may be 25ºC if 50% of the limiting species (Target-1, let's say) exists bound in Target-1, Probe-2. Further descriptions follow below.

NETTM_NAME=FirstNetTm;

The NetTm query is given a user defined name.

NETTM_FORMULA=Target-1,Probe-1

The complex (=species) to which the NetTM is applied. Therefore, the above formula defines the temperature at which at least some amount (usually 50%, set by NETTM_THRESHOLD, see below) of Target-1 is bound in the heterodimer (Target-1,Probe-1). Species within complexes are delimited with commas.

NETTM_THRESHOLD=50

The threshold of the NetTM is declared here. If a threshold is equal to 55, then the NETTM is defined as the temperature at which 55% of the limiting species (usually the least concentrated) is bound in a complex.

NETTM_LIMITING_SEQUENCE=Target-1

This is where the limiting species of a complex is defined. It is possible to say that a Target is at 50nM and a Probe is at 25nM and to declare the Target as the limiting species. This example would mean that the NetTm of Target,Probe heterodimer is defined as the temperature at which 50% (if the threshold is 50) of the Target is bound in the heterodimer.

2.2 General AOI2Server Input Files

2.2.1 Project Information Section

[project information]

Describes general information about the AOI2 design including identification and output conditions of the design.

NAME=ProbeDesign1

The name of the experiment.

DESCRIPTION=This experiment designs TaqMan probes against 4 targets.

A descriptor (optional) for the design. Not used by the AOI2Server engine. It is only meant as a header for users.

VERSION=7/31/2006

A version descriptor (optional) for the design. Not used by the AOI2Server engine. It is only meant as a header for users.

USER_NAME=JohnDoe

The (optional) name of the user. Not used by the AOI2Server engine. It is only meant as a header for users.

NUM_RETURN=3

The number of design solutions to be returned. If a singleplex probe is designed, then 3 probes will be returned when NUM_RETURN=3. If a primer pair is designed, and the NUM_RETURN=3, then 3 sets of primers (total of 6) will be returned

REQUIRE_SOLUTIONS=True

Normally, if no solutions could be found that pass all design criteria, no solutions will be returned to the user (default of REQUIRE_SOLUTIONS is False). However, solutions can be forced by setting REQUIRE_SOLUTIONS=True. Oligos that failed at least one of the design criteria will be returned in the .osf with a negative Q-score. Candidates that failed at a later stage in the design algorithm (e.g. passed relatively more tests) have a less negative score than ones that failed earlier on and these candidates with a less negative score will be ranked higher.

DEFAULT_NUM_REDUNDANT=100

The default NUM_REDUNDANT (see NUM_REDUNDANT keyword further down). These are default values and the NUM_REDUNDANT may be changed for any design.

DEFAULT_NUM_REDUNDANT_2=100

The default NUM_REDUNDANT_2 (see NUM_REDUNDANT_2 keyword further down). These are default values and the NUM_REDUNDANT_2 may be changed for any design.

FIXED_DEFAULT=false

A sequence can either be fixed (as on a microarray surface) or free (as in solution). If FIXED_DEFAULT is false, then the sequence is considered free. These are default values and the fixed condition may be changed for any sequence.

STRAND_DEFAULT=single

Sequences may be single or double stranded. These are default settings and each sequence may be identified as single or double stranded in the sequences section. Currently double strand functionality is not implemented.

OUTPUT_DIRECTORY=C:\OMPData

All output files can be found in this directory. If the directory is not specified or if the directory does not exist, output files can be found in the same directory as the application file (AOI2Server.exe)

2.2.2 Solution Conditions Section

[solution]

In this section the conditions of the solution can be set, like temperature, salt and buffer concentrations.

ASSAY_TEMPERATURE=37

Assay temperature is measured in degrees Celsius. The valid assay temperature range is from 0ºC to 100ºC. However, the most accurate values will be localized around an assay temperature of 50ºC. Designs with assay temperatures between 20-80ºC are still reliable however, tend to drop off in reliability outside of these ranges.

MAGNESIUM_CONCENTRATION=0.5

Magnesium concentration is measured in M. Mg concentration ranges from 0 to 1M are reliable, and AOI2Server can take higher values. However, reliability decreases with the increase of this salt.

SODIUM_CONCENTRATION=1

Sodium concentration is measured in M. Sodium concentration ranges from 0.04 to 1M are reliable, and AOI2Server can take higher values. However, reliability decreases with the increase of this salt.

GLYCEROL_CONCENTRATION=2

Concentration of the buffer glycerol, measured in M.

DMSO_CONCENTRATION=3

Concentration of the buffer DMSO (dimethyl sulfoxide), measured in M.

FORMAMIDE_CONCENTRATION=4

Concentration of the buffer formamide, measured in M.

TMAC_CONCENTRATION=1

Concentration of the salt TMAC (Tetramethyl ammonium chloride), measured in M.

BETAINE_CONCENTRATION=0.5

Concentration of the zwitter-ion Betaine, measured in M.

POLYMER_SALT=true

Polymer duplexes (greater than 16 base pairs) have a different salt dependence than oligomers due to end effects and counterion condensation effects. If this keyword is set to TRUE (which is recommended best practice), then the software applies the polymer salt correction to all duplexes that are longer than 16 basepairs, while shorter duplexes and unimolecular folds are unaffected.

SUBOPTIMAL_ENABLE=true

Besides the optimal structure of a species (= the most energetically favorable configuration with the lowest ΔG) AOI2Server is able to also calculate structures that can be present at a lower percentage and of which the thermodynamics are close to the optimal structure, called suboptimal structures.

The following keywords define the energy range between optimal and suboptimal structures and how many suboptimal structures AOI2Server will consider.

MAX_STRUCTURES_MONOMER=15

The maximum number of monomer suboptimal structures allowed. MAX_STRUCTURES_MONOMER is a hard limit and in the example, AOI2Server will not consider more than 15 monomer suboptimal structures.

MAX_STRUCTURES_HOMODIMER=10

The maximum number of homodimer suboptimal structures allowed. MAX_STRUCTURES_HOMODIMER is a hard limit and in the example, AOI2Server will not consider more than 10 homodimer suboptimal structures.

MAX_STRUCTURES_HETERODIMER=12

The maximum number of heterodimer suboptimal structures allowed. MAX_STRUCTURES_HETERODIMER is a hard limit and in the example, AOI2Server will not consider more than 12 heterodimer suboptimal structures.

SUBOPTIMALWINDOW_MONOMER=50

There are three ways to describe how suboptimal structures should be reported (SUBOPTIMALWINDOW, SUBOPTIMALDISTANCE, and SUBOPTIMALENERGY); it is recommended that only advanced users use more than one descriptor at a time. The SUBOPTIMALWINDOW descriptions calculate the amount of structures by percentage of the optimal structure's ΔG that AOI2Server should consider using traceback. In the example here, the suboptimal window for all monomers is set to 50%: if the Optimal (most likely structure) ΔG for the sequence is -5 kcal, then AOI2Server will consider all structures with a ΔG between -2.5 and –5 kcal.

SUBOPTIMALWINDOW_HOMODIMER=50

Similar to SUBOPTIMALWINDOW_MONOMER.

SUBOPTIMALWINDOW_HETERODIMER=100

Similar to SUBOPTIMALWINDOW_MONOMER. If no heterodimer exists, then the program will exit early.

SUBOPTIMALDISTANCE_MONOMER=3

There are three ways to describe how suboptimal structures should be reported (SUBOPTIMALWINDOW, SUBOPTIMALDISTANCE, and SUBOPTIMALENERGY); it is recommended that only advanced users use more than one descriptor at once. The distance function is a measure of similarity. It can be defined as the minimum number of insertions, deletions and substitutions required to convert structure A to structure B. A value input for this keyword enables the user to define the minimum difference between an optimal structure and the first suboptimal structure. In the example, the first MONOMER suboptimal structure will have a minimum distance function of 3 between it and the MONOMER optimal. Similarly, the second MONOMER suboptimal will have a minimum distance function of 3 between it and the first MONODIMER suboptimal, and so on.

SUBOPTIMALDISTANCE_HOMODIMER=3

Similar to SUBOPTIMALDISTANCE_MONOMER. If no homodimer exists, then the program will exit early.

SUBOPTIMALDISTANCE_HETERODIMER=3

Similar to SUBOPTIMALDISTANCE_MONOMER. If no heterodimer exists, then the program will exit early.

SUBOPTIMALENERGY_WINDOW_MONOMER=1

There are three ways to describe how suboptimal structures should be reported (SUBOPTIMALWINDOW, SUBOPTIMALDISTANCE, and SUBOPTIMALENERGY); it is recommended that only advanced users use more than one descriptor at once. SUBOPTIMALENERGY_WINDOW is the maximum energy in kcal/mole that the ΔG of a suboptimal structure can be away from the optimal structure at the ASSAY_TEMPERATURE. Therefore in the example, if an optimal structure for a target is -5.67 kcal, then OMP will return all suboptimal structures between -5.67 and -4.67 kcal.

SUBOPTIMALENERGY_WINDOW_HOMODIMER=1

Similar to SUBOPTIMALENERGY_WINDOW_MONOMER.

SUBOPTIMALENERGY_WINDOW_HETERODIMER=1

Similar to SUBOPTIMALENERGY_WINDOW_MONOMER.

ALLOW_MONOMER_EXTENSIBILITY=FALSE

When set to FALSE, candidate oligos that will form extensible hairpins (monomer secondary structures), as defined by the EXTENSION_OVERHANG, EXTENSION_WINDOW, and EXTENSION_MIN_PAIRS keywords (see definitions below), will be rejected as suitable candidates. When set to TRUE, these candidates will not be rejected based on this design criterion.

ALLOW_HOMODIMER_EXTENSIBILITY=FALSE

When set to FALSE, candidate oligos that will form extensible homodimers (self-dimers), as defined by the EXTENSION_OVERHANG, EXTENSION_WINDOW, and EXTENSION_MIN_PAIRS keywords (see definitions below), will be rejected as suitable candidates. When set to TRUE, these candidates will not be rejected based on this design criterion.

ALLOW_HETERODIMER_EXTENSIBILITY=FALSE

When set to FALSE, candidate oligos that will form extensible unintended heterodimers with the remainder of the target sequence or any other enabled sequence during design, will be rejected as suitable candidates. When set to TRUE, these candidates will not be rejected based on this design criterion. The definition of “extensible” is defined by the EXTENSION_OVERHANG, EXTENSION_WINDOW, and EXTENSION_MIN_PAIRS keywords (see definitions below).

EXTENSION_OVERHANG=1

In the following example, a monomer is shown to have an extension overhang of 1 (in red). Some polymerases with exonuclease activity do not require a blunt ended 3’ terminus, thus making the identification of species (monomer, homodimer, heterodimer etc.) with a certain overhang significant. When the keyword EXTENSION_OVERHANG is used, a species with an overhang on the 3’ end between 0 and the value of the keyword will be considered as “extensible” by the design algorithm. Setting EXTENSION_OVERHANG to 0 will only consider 3’ blunt ends extensible.
If this keyword isn’t specified, the default value of 1 will be used.

```
 ____________ACGTA   3’
/             ||||
\____________TGCACCGATGGAGTA   5’
```

EXTENSION_WINDOW=3

A minimum template length for a species to be considered “extensible” by polymerase can be specified with this keyword. In this example of value 3, 3 or more nucleotides need to be available on the strand to which the 3’ extensible end is hybridized. These nucleotides then function as the extension template.
If this keyword isn’t specified, the default value of 3 will be used.

EXTENSION_MIN_PAIRS=4

A minimum number of basepairs need to be formed between the oligo’s 3’ end and the template strand it hybridizes to. Matching basepairs and single internal mismatches are counted towards this keyword.
If this keyword isn’t specified, the default value of 4 will be used.

2.2.3 Blast Settings Section

[Blast Settings]

In this section the information necessary when BLAST is used during the design is specified, like the database to BLAST against and the word size.

RUN_BLAST=TRUE

When set to true, the candidate design sequences will be BLAST-ed against specified database (BLAST_DATABASE keyword, see below) as a last step of the design process.

BLAST_EXECUTABLE=C:\Program Files\DNASoftware\BLAST\Blastall.exe

The path and location of the BLAST executable file. Blastcl3.exe is the client executable. Blastall.exe is the normal BLAST executable.

BLAST_DATABASE=C:\OMPData\FASTAfiles\ecoli.fas

The database must be local or a legal online source. If local, the database must be in a formatted FASTA style. The FASTA file may contain more than one sequence entry. FASTA files may be formatted using formatdb.exe, usually found in the same folder as the blast executable. AOI2Server assumes that the database, if local is present and properly formatted. Legal online databases can be found at: http://www.ncbi.nlm.nih.gov/BLAST/blast_databases.shtml

BLAST_WORD_SIZE=11

The query sequence and every database sequence is split up into every possible "word" of a selected size. The default word size is 11 bp for DNA and must be greater or equal to 7 for DNA. Each word of a database sequence that is similar enough to a word in the query sequence is then tried to be extended in both directions, until the similarity between the two extended sequences is higher than a threshold. If you are interested in longer regions of homology you should increase the word size. Increasing the word size also speeds up the search, especially with larger query sequences *(>5kb)* and large databases.

DEFAULT_BLAST_STRAND_TYPE=BOTH

DEFAULT_BLAST_STRAND_TYPE is the DNA strand which is to be searched. The default value is “both” strands, other options are “ANTISENSE” or “SENSE”.

DEFAULT_BLAST_SEARCH_FOR=BINDING SITES

BLAST can search for binding sites or sequence similarity. AOI2Server is defaulted to look for "BINDING SITES".
DEFAULT_BLAST_SEARCH_FOR=SEQUENCE_SIMILARITY will perform a BLAST search which queries the sequence of interest for sequence similarity.

2.2.4 ThermoBlast Settings Section

[ThermoBlast Settings]

In this section the information necessary when ThermoBLAST is used during the design is specified, like the database to Blast against and the word size.
This option is only available if the ThermoBLAST module is licenses.

RUN_THERMOBLAST=TRUE

When set to true, the candidate design sequences will be ThermoBLAST-ed against specified database (THERMOBLAST_DATABASE keyword, see below) as a last step of the design process.

THERMOBLAST_EXECUTABLE=C:\Program Files\DNASoftware\ThermoBLAST\Blastn.exe

The path and location of the ThermoBLAST executable file.

THERMOBLAST_DATABASE=C:\OMPData\FASTAfiles\ecoli.fas

The database must be local or a legal online source. If local, the database must be in a formatted FASTA style. The FASTA file may contain more than one sequence entry. FASTA files may be formatted using formatdb.exe, usually found in the same folder as the blast executable. AOI2Server assumes that the database, if local is present and properly formatted. Legal online databases can be found at: http://www.ncbi.nlm.nih.gov/BLAST/blast_databases.shtml

2.2.5 Sequences Section

SEQUENCE_NAME=ATP7BGene

This is the name of the Sequence.

SEQUENCE=ATGTTGCTGCTGCTGCTGCTGCTGTGCTACTGCTGACACTCCTGAC

The sequence of the sequence specified by SEQUENCE_NAME.

SEQUENCE_TYPE=DNA

The nature of the sequence, e.g. DNA or RNA. PNA, Mor (morpholino), PSD (phosphorothioate), and OM2 (2'-O-Methyl) are also possible if the Modifieds module is licensed.

CONCENTRATION=0.001

The concentration in M of the sequence.

STRAND=SINGLE

Whether the sequence in single or double stranded. For oligos designed in AOI2Server, interactions between designed oligos and each strand of double stranded oligos (mishybridization) are considered. If set to Single, Probe type designs are only considered against the sense sequence; if set to Double, the probe can be designed against either strand. For Primer type designs, whether STRAND is set to SINGLE or DOUBLE doesn't influence the design: forward primers are automatically designed against the antisense strand and reverse primers automatically against the sense strand.

SEQUENCE_RANGE=100-300

The SEQUENCE_RANGE represents a subset of the total sequence. For example, it is possible to specify a 1000mer sequence (the Sequence keyword) and then only consider a smaller fragment for the design. In the above example, the sub-fragment would start at nucleotide 100 from the 5' end and end at nucleotide 300.

VARIATION_NAME=FIRSTVARIATION

The name of the variation.

VARIATION_POSITION=15

A variation to AOI2Server is defined as an insertion, deletion, or substitution mutation. The VARIATION_POSITION describes the first nucleotide of one of the variations above.

VARIATION_SEQUENCE=M

For substitution (S) variations, the VARIATION_SEQUENCE may be a IUPAC ambiguity code or an actual sequence (eg. TGTGA). For deletion (D) variations, the VARIATION_SEQUENCE value must be left blank. For insertion (I) variations, the base or sequence must be declared (e.g. AGGT).

VARIATION_TYPE=S

VARIATION_TYPE has three legal values: I, D, S. These letters represent insertion, deletion, and substitution respectively. Allele specific probes (design type of PROBE PAIR) can be designed against one of these variation types; AOI2Server will automatically create all combinations of wild type, mutant, sense and antisense strands.

VARIATION_LENGTH=0

The VARIATION_LENGTH represents the length of a VARIATION_TYPE. For all variations, the length may not be less than 0. Substitution and Insertion sequences are declared in VARIATION_SEQUENCE. The VARIATION_LENGTH of a substitution, if greater than 0, represents the length of the sequence to be replaced, not the length of the sequence that is being substituted in.

SEQUENCE_MODIFICATION_NAME=myMOD

The user defined name of the modification, modifications for this and the following 2 keywords are understood to refer to fluorophore and quencher labels.

SEQUENCE_MODIFICATION_TYPE=Alexa350

The desired quencher or fluor. The options are listed in the fluorquencher.txt file that can be found in the C:\Program Files\DNASoftware\OMP folder (for Visual OMP) or the folder where the executables (AOI2Server/AOI) are installed for OMP DE on Windows/Linux.

SEQUENCE_MODIFICATION_POSITION=4

The modification position defines the nucleotide that contains the specific modification.

2.2.6 Design Section

[Designs]

The design(s) of the AOI2Server experiment are listed in this section. Each design can use all or part of the keywords listed in this section.

OLIGO_NAME=MyProbeDesign

The name of the oligo being designed.

OLIGO_TYPE=PROBE

The nature of the oligo. The OLIGO_TYPE may be: PROBE, PROBE PAIR, PRIMER PAIR, FORWARD PRIMER, REVERSE PRIMER, ALLELE SPECIFIC PCR, FIXED PROBE, TAQMANPROBE, TAQMAN_MGB_PROBE, BEACON, or BEACON PAIR. Scorpion designs have a different format (.sif files instead of .odf) and are run by OMPServer, not by AOI2Server.

OLIGO_SEQUENCE_TYPE=RNA

The backbone type of the sequence, e.g. DNA or RNA. PNA, Mor (morpholino), PSD (phosphorothioate), and OM2 (2'-O-Methyl) are also possible if the Modifieds module is licensed.

OLIGO_TARGET_NAME=ATP7B

The name of sequence against which the oligo (probe or primer) is to be designed. The OLIGO_TARGET_NAME should match the SEQUENCE_NAME of one of the sequences specified in the Sequences section above.

OLIGO_CONCENTRATION=0.0001

The concentration in M of the oligo that is being designed.

OLIGO_ENABLED=TRUE

If an oligo is enabled (TRUE), then a design is performed. If an oligo is not enabled (FALSE), then a design is not performed. Therefore it is possible to list oligos and choose to perform or not perform designs for each listed design.

LNA_IN_PROBE_PAIR_DESIGN=TRUE

This keyword is only active if the Modifieds module is licensed also.

This keyword is only used for Allele Probe Pair design types (OLIGO_TYPE=PROBE PAIR). Besides the natural matching base, AOI2Server will also try to put an LNA modified base across from the SNP site. In general, LNA-DNA match basepairs are more stable than DNA-DNA match basepairs and LNA-DNA mismatches are more destabilizing than DNA-DNA match basepairs, therefore, placing an LNA nucleotide across a SNP will in general lead to better discrimination of the SNP.

OLIGO_PARAMETER_PROFILE_NAME=BALANCED

This is a keyword that acts as a comment. Typically Visual OMP may use this keyword in the GUI to determine a set of design parameters. On its own, used in the developer environment, it simply acts as a comment.

__Parameters to direct design to certain areas or to avoid certain regions__

OLIGO_TARGET_EXCLUDED_REGION=55,10

The OLIGO_TARGET_EXCLUDED_REGION describes a region on the sense strand of the target in which the probe may not be designed. The region is described by a start nucleotide (from the 5' end) and a length (in nucleotides); the values are separated by a comma. Therefore, in the example, AOI2Server will attempt to design a probe which does not contain nucleotides 55 to 64. This keyword can be used for primer design types also. Note, however that in the case of a primer pair, neither primer will be designed within the region, but both primers can be designed on either side of the region (so the region would be amplified).

Multiple excluded regions can be specified by repeating the keyword and setting the value to all desired regions that should be excluded.

RESTRICTED_MOTIF=GTAGTA

Any oligo design may not contain RESTRICTED_MOTIF. Therefore, in the example, AOI2Server will not consider oligos that contain "GTAGTA". The OLIGO_RESTRICTED_MOTIF may contain IUPAC ambiguity codes. Note that the RESTRICTED_MOTIF keyword can only be set after at least 1 design has been specified in the [Designs] section, but that it applies to all designs in the ODF file.

OLIGO_TARGET_RANGE=50-150

For probe type designs (PROBE, PROBE PAIR, FIXED PROBE, TAQMANPROBE, TAQMAN_MGB_PROBE, BEACON, or BEACON PAIR PROBE), this keyword indicates the region (counting with respect to the sense strand of the target) in which the probe needs to be designed; the all nucleotides of the probe will hybridize within this part of the target. In the example, the region starts at nucleotide number 50 from the 5' end and ends at nucleotide 150 from the 5' end of the target.

OLIGO_TARGET_AMPLIFICATION_WINDOW=50-150

The OLIGO_TARGET_AMPLIFICATION_WINDOW describes the area on the sense strand of the target that is to be amplified. Specifically, it is the region on the target around which primers are designed. Therefore in PCR primer design (PRIMER PAIR, REVERSE PRIMER, FORWARD PRIMER, or ALLELE SPECIFIC PCR design types), primers will be designed around nucleotides 50 through 150 from the 5' end of the sense strand of the Target, they will not hybridize within this region. In other words, counting with respect to the sense strand numbering: the forward primer will be designed downstream from nucleotide 50 (e.g. hybridizes in the target region of nucleotides 1-49), and the reverse primer will be designed upstream from nucleotide 150 (e.g. hybridizes in the target region of nucleotides 151-end of target).

OLIGO_MIN_AMPLIFICATION_WINDOW_SIZE=50

OLIGO_MIN_AMPLIFICATION_WINDOW_SIZE is defined as the minimum size in nucleotides of the area on the target that is to be amplified. Therefore , if a primer pair is being designed, then all combinations of forward and reverse primer candidates whose amplicon would be less than 50 nucleotides (in this example) would be discarded.

OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE=100

OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE is defined as the optimal size in nucleotides of the area on the target that is to be amplified. The OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE is a soft target: all amplicon sizes between the OLIGO_MIN_AMPLIFICATION_WINDOW_SIZE and OLIGO_MAX_AMPLIFICATION_WINDOW_SIZE will be accepted, but AOI2Server will prefer a combination of primers that will return an amplicon of this length (in the example, it is 100 nucleotides) based on the weights (OLIGO_AMPLIFICATION_WINDOW_SIZE_LT_WT, OLIGO_AMPLIFICATION_WINDOW_SIZE_GT_WT, and OLIGO_AMPLIFICATION_WINDOW_SIZE_SCALE).

OLIGO_MAX_AMPLIFICATION_WINDOW_SIZE=150

OLIGO_MAX_AMPLIFICATION_WINDOW_SIZE is defined as the maximum size in nucleotides of the area on the target that is to be amplified. Therefore , if a primer pair is being designed, then all combinations of forward and reverse primer candidates whose amplicon would be greater than 150 nucleotides (in this example) would be discarded.

OLIGO_AMPLIFICATION_WINDOW_SIZE_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on an oligo design per nucleotide that an amplicon is less than the optimal length. This weight is calculated for primers that create an amplicon of a length that is shorter than the OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE. Therefore if an OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE=100, then a primer set in which the amplicon is 99 nucleotides will be penalized 1 point. An amplicon length of 98 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which primers are ranked.

OLIGO_AMPLIFICATION_WINDOW_SIZE_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on an oligo design per nucleotide that an amplicon is greater than the optimal length. This weight is calculated for primers that create an amplicon of a length that is longer than the OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE. Therefore if an OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE=100, then an primer set in which the amplicon is 101 nucleotides will be penalized 1 point. An amplicon length of 102 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which primers are ranked.

OLIGO_AMPLIFICATION_WINDOW_SIZE_SCALE=LINEAR

The scale of OLIGO_AMPLIFICATION_WINDOW_SIZE_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the length of the amplicon is 5 nucleotides greater than the OLIGO_OPT_AMPLIFICATION_WINDOW_SIZE=100. In the example, this is 105 nucleotides. A square scale will give a 25 (5^2) point penalty to the same design.

Parameters that influence size of the solution space and extent of back-fill

SOLUTION_DISTANCE=5

The distance in nucleotides that one solution can be away from an adjacent solution. If the SOLUTION_DISTANCE=5, then AOI2Server will look for an oligo and then consider the next feasible oligo at least 5 nucleotides away (in this example). Therefore, the answers that are returned will not have high amounts of similarity and will be spread out over the target.

NUM_REDUNDANT=100

The number of candidate oligos (e.g probes or primers) that are passed from one heuristic to another is equal to the product of the NUM_RETURN and the NUM_REDUNDANT. The NUM_REDUNDANT is a keyword that is necessary to ensure that all solutions do or do not contain redundant solutions. Therefore, using default values, the top solutions will differ significantly from other solutions.

NUM_REDUNDANT_2=100

The number of candidate oligos (e.g probes or primers) that are passed from one heuristic to another AT THE POINT OF MISHYBRIDIZATION AND CROSSHYBRIDIZATION is equal to the product of the NUM_RETURN and the NUM_REDUNDANT_2. The NUM_REDUNDANT_2 is a keyword that is necessary to ensure that all solutions do or do not contain redundant solutions. Therefore, using default values, the top solutions will differ significantly from other solutions.

Parameters to specify variations on a sequence

OLIGO_TARGET_VAR_NAME

The name of the variation (usually a SNP) on the target that is to be detected by the allele specific design. Used for PROBE PAIR, ALLELE SPECIFIC PCR, and BEACON PAIR design types (OLIGO_TYPE keyword). This value needs to match the VARIATION_NAME of 1 of the variations specified for the target sequence in the [Sequences] section.

OLIGO_TARGET_VAR_POS

The position of the variation on the Target as determined from the 5' terminus. Used mostly when describing a location for a SNP but may also be used to describe the start position of an insertion or deletion variation. Value needs to match the VARIATION_POSITION of 1 of the variations specified for the target sequence in the [Sequences] section.

OLIGO_MIN_VAR_POS=3

The minimum nucleotide from the 5' end of the probe which will assay for the variation. Therefore, in the example, if a probe pair is being designed, then the position on the probe that is assaying a SNP may not be within 3 nucleotides of the 5' end of the probe.

OLIGO_OPT_VAR_POS=LENGTH/2

OLIGO_OPT_VAR_POS refers to the optimal location for an assaying position on a probe. Therefore, if a probe pair is being designed against a SNP, in the example, it is indicated that the best location is the middle of the probe (LENGTH/2), the keywords LENGTH indicates the last nucleotide of a design, e.g. the 3' end nucleotide.

OLIGO_MAX_VAR_POS=LENGTH-2

The last position on a probe that may assay for a variation is defined as the OLIGO_MAX_VAR_POS. The default is set to 3 nucleotides from the end of the probe (LENGTH-2). Therefore, the assaying position cannot be within 3 nucleotides from the 3' terminus.

OLIGO_VAR_POS_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on primers per nucleotide that a SNP assaying position is LESS than the optimal location. Therefore if an OLIGO_OPT_VAR_POS=10, the candidate probe length=20, and the design of the candidate probe has the SNP assayed at position 6, then this candidate will receive a penalty of 4 points.

These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_VAR_POS_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on primers per nucleotide that a SNP assaying position is GREATER than the optimal location. Therefore if an OLIGO_OPT_VAR_POS=10, the candidate probe length=20, and the design of the candidate probe has the SNP assayed at position 14, then this candidate will receive a penalty of 4 points.

These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_VAR_POS_WT_SCALE=LINEAR

The scale of OLIGO_VAR_POS_WT may be linear or square. Therefore, using the above 4 keywords, the weights or penalties assessed to designs in which the assaying position is farther away from an optimal position may be linear or exponential. A linear scale will penalize solutions with the SNP assaying position 5 nucleotides away from the optimal position 5 points; a square scale will penalize the same design 25 points.

Parameters to set the length of the oligos being designed

OLIGO_MIN_SIZE=15

OLIGO_MIN_SIZE is defined as the minimum size in nucleotides of each oligo (primer or probe). OLIGO_MIN_SIZE is a hard limit and in the example, AOI2Server will not consider oligos less than 15 nucleotides.

OLIGO_OPT_SIZE=25

OLIGO_OPT_SIZE is defined as the optimal size in nucleotides of each oligo (primer or probe). The OLIGO_OPT_SIZE is a soft target; AOI2Server will favor oligos of this length (in the example, it is 25 nucleotides) over oligos of other lengths.

OLIGO_MAX_SIZE=30

OLIGO_MAX_SIZE is defined as the maximum size in nucleotides of each oligo (primer or probe). OLIGO_MAX_SIZE is a hard limit and in the example, AOI2Server will not consider oligos greater than 30 nucleotides.

OLIGO_SIZE_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per nucleotide that an oligo is less than the optimal length. This weight is calculated for oligos that are less than the OLIGO_OPT_SIZE. Therefore if an OLIGO_OPT_SIZE=25, then a 24 nucleotide oligo will be penalized 1 point, a 23 nucleotide oligo will be penalized 2 points, and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_SIZE_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per nucleotide that an oligo is greater than the optimal length. This weight is calculated for oligos that are greater than the OLIGO_OPT_SIZE. Therefore if an OLIGO_OPT_SIZE=25, then a 26 nucleotide oligo will be penalized 1 point, a 27 nucleotide oligo will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_SIZE_WT_SCALE=LINEAR

The scale of OLIGO_SIZE_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the length of the oligo is 5 nucleotides greater than the OLIGO_OPT_SIZE. In the example, this is 30 nucleotides. A square scale will give a 25 point penalty to the same design.

Parameters to set the desired Tm and dG of the intended target-oligo duplex

OLIGO_MIN_TM=55

OLIGO_MIN_TM is defined as the minimum Tm in ℃ of each oligo (primer or probe) hybridized to the target it is being designed for. OLIGO_MIN_TM is a hard limit and in the example, AOI2Server will not consider oligos with a Tm less than 55 nucleotides.

OLIGO_OPT_TM=60

OLIGO_OPT_TM is defined as the optimal Tm in ℃ of each oligo (primer or probe) hybridized to the target it is being designed for. The OLIGO_OPT_TM is a soft target; AOI2Server will first prefer oligos of this Tm (in the example, it is 60℃) over ones with other Tm's.

OLIGO_MAX_TM=80

OLIGO_MAX_TM is defined as the maximum Tm in ℃ of each oligo (primer or probe) hybridized to the target it is being designed for. OLIGO_MAX_TM is a hard limit and in the example, AOI2Server will not consider oligos with a Tm greater than 80℃.

OLIGO_TM_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that the Tm of an oligo-target duplex is less than the optimal Tm. This weight is calculated for deuplexes that are less than the OLIGO_OPT_TM. Therefore if an OLIGO_OPT_TM=60, then a duplex with a Tm of 59 will be penalized 1 point, a Tm of 58 will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_TM_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that the Tm of a duplex is greater than the optimal Tm. This weight is calculated for duplexes that are greater than the OLIGO_OPT_TM. Therefore if an OLIGO_OPT_TM=60, then a duplex with a Tm of 61 will be penalized 1 point, a Tm of 62 will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_TM_WT_SCALE=LINEAR

The scale of OLIGO_TM_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the Tm of the duplex is 5℃ greater than the OLIGO_OPT_TM. In the example, this is 65℃. A squ are scale will give a 25 point penalty to the same design.

OLIGO_MIN_DG=-100

OLIGO_MIN_DG is defined as the minimum dG in kcal/mole of each oligo (primer or probe) hybridized to the target. OLIGO_MIN_DG is a hard limit and in the example, AOI2Server will not consider oligos less (more negative) than -100 kcal/mole.

OLIGO_OPT_DG=-24

OLIGO_OPT_DG is defined as the optimal dG in kcal/mole of each oligo (primer or probe) hybridized to the target. The OLIGO_OPT_DG is a soft target; AOI2Server will prefer oligos of this kcal/mole (in the example, it is -24 kcal/mol) before considering other dG's.

OLIGO_MAX_DG=-3

OLIGO_MAX_DG is defined as the maximum dG in kcal/mole of each oligo (primer or probe) hybridized to the target. OLIGO_MAX_DG is a hard limit and in the example, AOI2Server will not consider oligos greater (more positive) than -3 kcal/mole.

OLIGO_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that a target-oligo duplex is less (more negative) than the optimal dG. This weight is calculated for duplexes that are less than the OLIGO_OPT_DG. Therefore if an OLIGO_OPT_DG=-24, then a duplex with a dG of -25 kcal/mole will be penalized 1 point, a duplex with a dG of -26 kcal/mole will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_DG_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on duplexes per kcal/mole that a duplex is greater (more positive) than the optimal dG. This weight is calculated for duplexes that are less than the OLIGO_OPT_DG. Therefore if an OLIGO_OPT_DG=-24, then a duplex with a dG of -23 kcal/mole will be penalized 1 point, a duplex with a dG of -22 kcal/mole will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_DG_WT_SCALE=LINEAR

The scale of OLIGO_DG_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the dG of the oligo-target duplex is 5 kcal/mole greater than the OLIGO_OPT_DG. In the example, this is -19 kcal/mole. A square scale will give a 25 point penalty to the same design.

Parameters to set the desired GC content of the oligo being designed

OLIGO_MIN_GCC=30

OLIGO_MIN_GCC is defined as the minimum GC content in percentage of each oligo (primer or probe). OLIGO_MIN_GCC is a hard limit and in the example, AOI2Server will not consider oligos with a GC content of less than 30%.

OLIGO_OPT_GCC=55

OLIGO_OPT_GCC is defined as the optimal GC content in percentage of each oligo (primer or probe). The OLIGO_OPT_GCC is a soft target; AOI2Server will prefer oligos of this GC content (in the example, it is 55%) over oligos with other GC percentages.

OLIGO_MAX_GCC=80

OLIGO_MIN_GCC is defined as the maximum GC content in percentage of each oligo (primer or probe). OLIGO_MAX_GCC is a hard limit and in the example, AOI2Server will not consider oligos with a GC content of greater than 80%.

OLIGO_GCC_LT_WT

This is the relative penalty (weight) that AOI2Server places on oligos per percentage that an oligo is less than the optimal GC content percentage. This weight is calculated for oligos that are less than the OLIGO_OPT_GCC. Therefore if an OLIGO_OPT_GCC=55, then a oligo with a GC content of 54% will be penalized 1 point, and an oligo with a GC content of 53% will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_GCC_GT_WT

This is the relative penalty (weight) that AOI2Server places on oligos per percentage that an oligo is greater than the optimal GC content percentage. This weight is calculated for oligos that are greater than the OLIGO_OPT_GCC. Therefore if an OLIGO_OPT_GCC=55, then a oligo with a GC content of 56% will be penalized 1 point, and an oligo with a GC content of 57% will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_GCC_WT_SCALE=LINEAR

The scale of OLIGO_GCC_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the GC content in % of the oligo is 5% greater than the OLIGO_OPT_GCC. In the example, this is 60%. A square scale will give a 25 point penalty to the same design.

Parameters to set parameters that prevent oligos from being designed in target regions with considerable secondary structure

OLIGO_TARGET_LOCAL_LENGTH=35

When a probe or primer is designed against a target, AOI2Server automatically folds a short segment of the target to determine the thermodynamics of the secondary structure of the local target. The length of the local target segment is set using OLIGO_TARGET_LOCAL_LENGTH.

OLIGO_OPT_TARGET_LOCAL_TM=0

OLIGO_OPT_TARGET_LOCAL_TM is defined as the optimal Tm in ℃ of the segment of the target to be checked for secondary structure. The OLIGO_OPT_TARGET_LOCAL_TM is a soft target; AOI2Server will prefer oligos in segments of the target with this Tm (in the example, it is 0℃) over other oligos.

OLIGO_MAX_TARGET_LOCAL_TM=65

OLIGO_MAX_TARGET_LOCAL_TM is defined as the maximum Tm in ℃ of the local target segment. OLIGO_MAX_TARGET_LOCAL_TM is a hard limit and in the example, AOI2Server will not consider target segments with a Tm greater than 65℃.

OLIGO_TARGET_LOCAL_TM_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that a TARGET is greater than the optimal TM. This weight is calculated for oligos that are greater than the OLIGO_OPT_TARGET_LOCAL_TM. Therefore if an OLIGO_OPT_TARGET_LOCAL_TM=55, then a oligo design will be penalized 1 point if the corresponding folded target segment has a Tm of 56 and will be penalized 2 points if the target has a Tm of 57 and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_TARGET_LOCAL_TM_WT_SCALE=LINEAR

The scale of OLIGO_TARGET_LOCAL_TM_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to an oligo design where the Tm of the corresponding Target is 5℃ greater than the OLIGO_O PT_TARGET_LOCAL_TM. In the example, this is 60℃. A square scale will give a 25 point penalty to the same design.

OLIGO_OPT_TARGET_LOCAL_DG=0

OLIGO_OPT_TARGET_LOCAL_DG is defined as the optimal dG in kcal/mole of the segment of the target to be checked for secondary structure. The OLIGO_OPT_TARGET_LOCAL_DG is a soft target; AOI2Server will prefer oligos in segments of the target of this dG (in the example, it is 0 kcal/mol) over other oligos.

OLIGO_MIN_TARGET_LOCAL_DG=-50

OLIGO_MIN_TARGET_LOCAL_DG is defined as the minimum dG in kcal/mole of the local target segment. OLIGO_MIN_TARGET_LOCAL_DG is a hard limit and in the example, AOI2Server will not consider target segments with a dG less than -50 kcal/mole.

OLIGO_TARGET_LOCAL_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that a target segment is less than the optimal dG. This weight is calculated for segments that are less than the OLIGO_OPT_TARGET_LOCAL_DG. Therefore if an OLIGO_OPT_TARGET_LOCAL_DG=0, then a oligo design will be penalized 1 point if the corresponding folded target segment has a dG of -1 kcal/mole and will be penalized 2 points if the target has a dG of -2 kcal/mole and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_TARGET_LOCAL_DG_WT_SCALE=LINEAR

The scale of OLIGO_TARGET_LOCAL_DG_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to an oligo design where the Tm of the corresponding target segment is 5 kcal greater than the OLIGO_OPT_TARGET_LOCAL_TM. In the example, this is +5 kcal/mole. A square scale will give a 25 point penalty to the same design.

Parameters to prevent designs of oligos that form strong hairpins

OLIGO_OPT_MONOMER_TM=2

OLIGO_OPT_MONOMER_TM is defined as the optimal Tm of the folded oligo (primer or probe) . The OLIGO_OPT_MONOMER_TM is a soft target; AOI2Server will prefer oligos of this Tm (in the example, it is 2℃) over oligos of other Tm's.

OLIGO_MAX_MONOMER_TM=65

OLIGO_MAX_MONOMER_TM is defined as the maximum Tm of the folded oligo (primer or probe). OLIGO_MAX_MONOMER_TM is a hard limit and in the example, AOI2Server will not consider oligos whose folded conformation has a Tm greater than 65℃.

OLIGO_MONOMER_TM_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that an oligo is greater than the optimal Tm in a folded conformation. This weight is calculated for oligos that are greater than the OLIGO_OPT_MONOMER_TM. Therefore if an OLIGO_OPT_MONOMER_TM=2, then an oligo with Tm of 3°C will be penalized 1 point, an oligo with a Tm of 4℃ will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_MONOMER_TM_WT_SCALE=LINEAR

The scale of OLIGO_MONOMER_TM_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the Tm of the folded conformation of the oligo is 5℃ greater than the OLIGO_OPT_MONOMER_TM. In the example, this is 7℃. A square scale will give a 25 point penalty to the same design.

OLIGO_MIN_MONOMER_DG=-50

OLIGO_MIN_MONOMER_DG is defined as the minimum dG in kcal/mole of the folded oligo (primer or probe). OLIGO_MIN_MONOMER_DG is a hard limit and in the example, AOI2Server will not consider oligos whose folded conformation has a dG less than -50 kcal/mole.

OLIGO_OPT_MONOMER_DG=0

OLIGO_OPT_MONOMER_DG is defined as the optimal dG of the folded oligo (primer or probe). The OLIGO_OPT_MONOMER_DG is a soft target; AOI2Server will prefer oligos of this dG (in the example, it is 0 kcal/mol) over oligos of other dG's.

OLIGO_MONOMER_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that an oligo is less than the optimal dG in a folded conformation. This weight is calculated for oligos that are less than the OLIGO_OPT_MONOMER_DG. Therefore if an OLIGO_OPT_MONOMER_DG=0, then an oligo with dG of -1 kcal/mole will be penalized 1 point, an oligo with a dG of -2 kcal/mole will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_MONOMER_DG_WT_SCALE=LINEAR

The scale of OLIGO_MONOMER_DG_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the dG of the folded conformation of the oligo is 5 kcal/mole greater than the OLIGO_OPT_MONOMER_DG. In the example, this is 5 kcal/mole. A square scale will give a 25 point penalty to the same design.

Parameters to prevent oligos to form strong homodimers (self dimers)

OLIGO_OPT_HOMODIMER_TM=2

OLIGO_OPT_HOMODIMER_TM is defined as the optimal Tm of the homodimer of the oligo (primer or probe). The OLIGO_OPT_HOMODIMER_TM is a soft target; AOI2Server will prefer oligos with homodimers of this Tm (in the example, it is 2℃) over oligos with homodimers of other Tm's.

OLIGO_MAX_HOMODIMER_TM=65

OLIGO_MAX_HOMODIMER_TM is defined as the maximum Tm of the homodimer of the oligo (primer or probe). OLIGO_MAX_HOMODIMER_TM is a hard limit and in the example, AOI2Server will not consider oligos whose homodimers have a Tm greater than 65℃.

OLIGO_HOMODIMER_TM_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that the Tm of the homodimer of the oligo is greater than the optimal homodimer Tm. This weight is calculated for oligos that are greater than the OLIGO_OPT_HOMODIMER_TM. Therefore if an OLIGO_OPT_HOMODIMER_TM=2, then an oligo with a homodimer Tm of 3℃ will be penalized 1 point, an oligo with a ho modimer Tm of 4℃ will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_HOMODIMER_TM_WT_SCALE=LINEAR

The scale of OLIGO_HOMODIMER_TM_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the Tm of homodimer of the oligo is 5℃ greater than the OLIGO_OPT_HOMODIMER_TM. In th e example, this is 7℃. A square scale will give a 25 point penalty to the same design.

OLIGO_MIN_HOMODIMER_DG=-50

OLIGO_MIN_HOMODIMER_DG is defined as the minimum dG in kcal/mole of the homodimer of the oligo (primer or probe). OLIGO_MIN_HOMODIMER_dG is a hard limit and in the example, AOI2Server will not consider oligos whose homodimers have a dG less than -50 kcal/mole.

OLIGO_OPT_HOMODIMER_DG=0

OLIGO_OPT_HOMODIMER_DG is defined as the optimal dG in kcal/mole of the homodimer of the oligo (primer or probe). The OLIGO_OPT_HOMODIMER_DG is a soft target; AOI2Server will prefer oligos with homodimers of this dG (in the example, it is 0 kcal/mol) over oligos with homodimers of other dG's.

OLIGO_HOMODIMER_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that the dG of the homodimer of the oligo is less than the optimal homodimer dG. This weight is calculated for oligos that are less than the OLIGO_OPT_HOMODIMER_DG. Therefore if an OLIGO_OPT_HOMODIMER_DG=0, then an oligo with a homodimer dG of -1 kcal/mole will be penalized 1 point, an oligo with a homodimer dG of -2 kcal/mole will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_HOMODIMER_DG_WT_SCALE=LINEAR

The scale of OLIGO_HOMODIMER_DG_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the dG of homodimer of the oligo is 5 kcal/mole greater than the OLIGO_OPT_HOMODIMER_DG. In the example, this is 5 kcal/mole. A square scale will give a 25 point penalty to the same design.

Parameters to prevent the oligo from strong interactions with unintended targets

OLIGO_OPT_MISHYB_TM=2

OLIGO_OPT_MISHYB_TM is defined as the optimal Tm of any mishybridization that the oligo (primer or probe) has with any sequence listed at the top of the ODF including a mishybridization on the intended target at a position other than the design location. The OLIGO_OPT_MISHYB_TM is a soft target; AOI2Server will prefer oligos with mishybridizations at this Tm (in the example, it is 2℃) over oligos with mishybridizations at other Tm's.

OLIGO_MAX_MISHYB_TM=45

OLIGO_MAX_MISHYB_TM is defined as the maximum Tm of any mishybridization that the oligo (primer or probe) has with any sequence listed at the top of the ODF including a mishybridization on the intended target at a position other than the design location. The OLIGO_MAX_MISHYB_TM is a hard limit and in the example, AOI2Server will not consider oligos whose mishybridizations have a Tm greater than 45℃.

OLIGO_MISHYB_TM_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that the Tm of any mishybridization is greater than the optimal mishybridization Tm. This weight is calculated for oligos that are less than the OLIGO_OPT_MISHYB_TM. Therefore if an OLIGO_OPT_MISHYB_TM=2, then an oligo with a mishybridization Tm of 3℃ will be penalized 1 point, an oligo with a mishybridization Tm of 4℃ will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_MISHYB_TM_WT_SCALE=LINEAR

The scale of OLIGO_MISHYB_TM_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the Tm of the mishybridization of the oligo is 5℃ greater than the OLIGO_OPT_MISHYB_TM. In the e xample, this is 7℃. A square scale will give a 25 point penalty to the same design.

OLIGO_MIN_MISHYB_DG=-50

OLIGO_MIN_MISHYB_DG is defined as the minimum dG in kcal/mole of any mishybridization that the oligo (primer or probe) has with any sequence listed at the top of the ODF including a mishybridization on the intended target at a position other than the design location. The OLIGO_MIN_MISHYB_DG is a hard limit and in the example, AOI2Server will not consider oligos whose mishybridizations have a dG less than -50.

OLIGO_OPT_MISHYB_DG=0

OLIGO_OPT_MISHYB_DG is defined as the optimal dG in kcal/mole of any mishybridization that the oligo (primer or probe) has with any sequence listed at the top of the ODF including a mishybridization on the intended target at a position other than the design location. The OLIGO_OPT_MISHYB_DG is a soft target; AOI2Server will prefer oligos with mishybridizations at this dG (in the example, it is 0 kcal/mol) over oligos with mishybridizations at other dG's.

OLIGO_MISHYB_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that the dG of any mishybridization is less than the optimal mishybridization dG. This weight is calculated for oligos that are less than the OLIGO_OPT_MISHYB_DG. Therefore if an OLIGO_OPT_MISHYB_DG=0, then an oligo with a mishybridization dG of -1 kcal/mole will be penalized 1 point, an oligo with a mishybridization dG of -2 kcal/mole will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_MISHYB_DG_WT_SCALE=LINEAR

The scale of OLIGO_MISHYB_DG_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the dG of the mishybridization of the oligo is 5 kcal/mole greater than the OLIGO_OPT_MISHYB_DG. In the example, this is +5 kcal/mole. A square scale will give a 25 point penalty to the same design.

Parameters to prevent the oligo from strong interactions with other oligos

OLIGO_OPT_CROSSHYB_TM=2

OLIGO_OPT_CROSSHYB_TM is defined as the optimal Tm of any crosshybridization that the oligo (primer or probe) has with any other oligo (probe or primer) designed at the same time. The OLIGO_OPT_CROSSHYB_TM is a soft target; AOI2Server will prefer oligos with crosshybridizations at this Tm (in the example, it is 2℃) over oligos with crosshybridizations at other Tm's.

OLIGO_MAX_CROSSHYB_TM=45

OLIGO_MAX_CROSSHYB_TM is defined as the maximum Tm of any crosshybridization that the oligo (primer or probe) has with any other oligo (probe or primer) designed at the same time. The OLIGO_MAX_CROSSHYB_TM is a hard limit and in the example, AOI2Server will not consider oligos whose crosshybridizations have a Tm greater than 45℃.

OLIGO_CROSSHYB_TM_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that the Tm of any crosshybridization is greater than the optimal crosshybridization Tm. This weight is calculated for oligos that are less than the OLIGO_OPT_CROSSHYB_TM. Therefore if an OLIGO_OPT_CROSSHYB_TM=2, then an oligo with a crosshybridization Tm of 3℃ will be penalized 1 point, an oligo with a crosshybridization Tm of 4℃ will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_CROSSHYB_TM_WT_SCALE=LINEAR

The scale of OLIGO_CROSSHYB_TM_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the Tm of the crosshybridization of the oligo is 5℃ greater than the OLIGO_OPT_CROSSHYB_TM . In the example, this is 7℃. A square scale will give a 25 point penalty to the same design.

OLIGO_MIN_CROSSHYB_DG=-50

OLIGO_MIN_CROSSHYB_DG is defined as the minimum dG in kcal/mole of any crosshybridization that the oligo (primer or probe) has with any other oligo (probe or primer) designed at the same time. The OLIGO_MIN_CROSSHYB_DG is a hard limit and in the example, AOI2Server will not consider oligos whose crosshybridizations have a dG less than -50.

OLIGO_OPT_CROSSHYB_DG=0

OLIGO_OPT_CROSSHYB_DG is defined as the optimal dG in kcal/mole of any crosshybridization that the oligo (primer or probe) has with any other oligo (probe or primer) designed at the same time. The OLIGO_OPT_CROSSHYB_DG is a soft target; AOI2Server will prefer oligos with crosshybridizations at this dG (in the example, it is 0 kcal/mol) over oligos with crosshybridizations at other dG's.

OLIGO_CROSSHYB_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that the dG of any crosshybridization is less than the optimal crosshybridization dG. This weight is calculated for oligos that are less than the OLIGO_OPT_CROSSHYB_DG. Therefore if an OLIGO_OPT_CROSSHYB_DG=0, then an oligo with a crosshybridization dG of -1 kcal/mole will be penalized 1 point, an oligo with a crosshybridization dG of -2 kcal/mole will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_CROSSHYB_DG_WT_SCALE=LINEAR

The scale of OLIGO_CROSSHYB_DG_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the dG of the crosshybridization of the oligo is 5 kcal/mole greater than the OLIGO_OPT_CROSSHYB_DG. In the example, this is +5 kcal/mole. A square scale will give a 25 point penalty to the same design.

OLIGO_MAX_POLY_N=4

The maximum number of consecutive standard bases to be allowed in an oligo (probe or primer) design. Therefore, in the example, an oligo will not be designed such that the oligo contains any string of AAAAA,CCCCC,GGGGG,TTTTT or more.

Parameters to prevent runs of equal bases

OLIGO_MAX_POLY_G=4

The maximum number of consecutive G's to be allowed in an oligo (probe or primer) design. Therefore, in the example, an oligo will not be designed such that the oligo contains any string of GGGGG or more.

OLIGO_MAX_POLY_C=4

The maximum number of consecutive C's to be allowed in an oligo (probe or primer) design. Therefore, in the example, an oligo will not be designed such that the oligo contains any string of CCCCC or more.

OLIGO_MAX_POLY_A=4

The maximum number of consecutive A's to be allowed in an oligo (probe or primer) design. Therefore, in the example, an oligo will not be designed such that the oligo contains any string of AAAAA or more.

OLIGO_MAX_POLY_T=4

The maximum number of consecutive T's to be allowed in an oligo (probe or primer) design. Therefore, in the example, an oligo will not be designed such that the oligo contains any string of TTTTT or more.

Parameters to optimize discrimination for allele specific designs (probe pair, allele specific PCR, and Beacon Probe Pair)

OLIGO_ALLELE_MIN_LENGTH_DIFFERENCE=1

OLIGO_ALLELE_MIN_LENGTH_DIFFERENCE is defined as the minimum difference in length (in nucleotides) between the two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_MIN_LENGTH_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos less different than 1 nucleotide in length.

OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE=2

OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE is defined as the optimal difference in length (in nucleotides) between the two probes (one against wild type, the other against the mutant) The OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE is a soft target; AOI2Server will prefer oligos of this difference in length (in the example, it is 2 nucleotides) over oligos with other differences in lengths.

OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE=5

OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE is defined as the maximum difference in length (in nucleotides) between the two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos more different in length than 8 nucleotides.

OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on a design per nucleotide that the difference between one probe (e.g. wild type) and another probe (e.g. mutant) is less than the OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE. Therefore if an OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE=2, then a difference in length between the probes of 1 will be penalized 1 point, a difference of 2 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on a design per nucleotide that the difference between one probe (e.g. wild type) and another probe (e.g. mutant) is greater than the OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE. Therefore if an OLIGO_ALLELE_OPT_LENGTH_DIFFERENCE=2, then a difference in length between the probes of 3 will be penalized 1 point, a difference of 4 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE_WT_SCALE=LINEAR

The scale of OLIGO_ALLELE_MAX_LENGTH_DIFFERENCE_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the length of one probe in a probe pair is 5 nucleotides longer than the length of the other probe in the same probe pair. In the example, this is 5 nucleotides. A square scale will give a 25 point penalty to the same design.

OLIGO_ALLELE_MIN_VAR_POS_DIFFERENCE=1

OLIGO_ALLELE_MIN_VAR_POS_DIFFERENCE is defined as the minimum length in nucleotides between the variation assay positions of two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_MIN_VAR_POS_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos less different in variation assay position than 1 nucleotide.

OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE=2

OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE is defined as the optimal difference between the variation assay positions of two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE is a soft target; AOI2Server will prefer oligos with this variation position difference (in the example, it is 2 nucleotides) over oligos with other variation position differences.

OLIGO_ALLELE_MAX_VAR_POS_DIFFERENCE=5

OLIGO_ALLELE_MAX_VAR_POS_DIFFERENCE is defined as the maximum length in nucleotides between the variation assay positions of two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_MAX_VAR_POS_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos more different in variation assay position than 5 nucleotides.

OLIGO_ALLELE_VAR_POS_DIFFERENCE_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on a design per nucleotide that the difference between the variation position on one probe (e.g. wild type) and the variation position on another probe (e.g. mutant) is less than the OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE. Therefore if an OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE=2, then a difference between the variation positions on probes of 1 will be penalized 1 point, a difference of 2 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_ALLELE_VAR_POS_DIFFERENCE_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on a design per nucleotide that the difference between the variation position on one probe (e.g. wild type) and the variation position on another probe (e.g. mutant) is greater than the OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE. Therefore if an OLIGO_ALLELE_OPT_VAR_POS_DIFFERENCE=2, then a difference between the variation positions on probes of 3 will be penalized 1 point, a difference of 4 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_ALLELE_VAR_POS_DIFFERENCE_WT_SCALE=LINEAR

The scale of OLIGO_ALLELE_VAR_POS_DIFFERENCE_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the difference between the variation positions of two probes in a probe pair is 5 nucleotides. In the example, this is 5 nucleotides. A square scale will give a 25 point penalty to the same design.

OLIGO_ALLELE_MIN_START_POS_DIFFERENCE=1

OLIGO_ALLELE_MIN_START_POS_DIFFERENCE is defined as the minimum length in nucleotides between the start positions of two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_MIN_START_POS_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos with start positions less than 1 nucleotide apart.

OLIGO_ALLELE_OPT_START_POS_DIFFERENCE=2

OLIGO_ALLELE_OPT_START_POS_DIFFERENCE is defined as the optimal difference between the start positions of two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_OPT_START_POS_DIFFERENCE is a soft target; AOI2Server will prefer oligos of this difference in start position (in the example, it is 2 nucleotides) over oligos with other differences in start position.

OLIGO_ALLELE_MAX_START_POS_DIFFERENCE=5

OLIGO_ALLELE_MAX_START_POS_DIFFERENCE is defined as the maximum length in nucleotides between the start positions of two probes (one against wild type, the other against the mutant) in a probe pair. OLIGO_ALLELE_MAX_START_POS_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos with start positions more than 5 nucleotides apart.

OLIGO_ALLELE_START_POS_DIFFERENCE_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on a design per nucleotide that the difference between the start position on one probe (e.g. wild type) and the start position on another probe (e.g. mutant) is less than the OLIGO_ALLELE_OPT_START_POS_DIFFERENCE. Therefore if an OLIGO_ALLELE_OPT_START_POS_DIFFERENCE=2, then a difference between the start positions on probes of 1 will be penalized 1 point, a difference of 2 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_ALLELE_START_POS_DIFFERENCE_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on a design per nucleotide that the difference between the start position on one probe (e.g. wild type) and the start position on another probe (e.g. mutant) is greater than the OLIGO_ALLELE_OPT_START_POS_DIFFERENCE. Therefore if an OLIGO_ALLELE_OPT_START_POS_DIFFERENCE=2, then a difference between the start positions on probes of 3 will be penalized 1 point, a difference of 4 nucleotides will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_ALLELE_START_POS_DIFFERENCE_WT_SCALE=LINEAR

The scale of OLIGO_ALLELE_START_POS_DIFFERENCE_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the difference between the start positions of two probes in a probe pair is 5 nucleotides. In the example, this is 5 nucleotides. A square scale will give a 25 point penalty to the same design.

OLIGO_MIN_TM_ALLELE_DIFFERENCE=3

When designing a PROBE PAIR (or ALLELE SPECIFIC PCR or BEACON PAIR) it is necessary to quantify the minimum amount that the two probes/primers (wild type and mutant) must differ from each other. This can be done by Tm or dG. The example shows that a mutant probe/primer must bind to the wild type target with a Tm at least 3 degrees less than the hybridization with the mutant target, and vice versa. The OLIGO_MIN_TM_ALLELE_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos whose allelic mishybridizations have a difference in Tm less than 3℃.

OLIGO_OPT_TM_ALLELE_DIFFERENCE=10

OLIGO_OPT_TM_ALLELE_DIFFERENCE is defined as the optimal Tm that one probe (e.g. wild type) can mishybridize against the other target allele (e.g. mutant). The OLIGO_OPT_TM_ALLELE_DIFFERENCE is a soft target; AOI2Server will prefer oligos with these allelic mishybridization differences in Tm (in the example, it is 10℃) over oligos with other allelic mihybridization differences in Tm.

OLIGO_TM_ALLELE_DIFFERENCE_WT=3

This is the relative penalty (weight) that AOI2Server places on oligos per ℃ that the Tm difference between 2 probes is less or greater than OLIGO_OPT_TM_ALLELE_DIFFERENCE. Therefore if an OLIGO_OPT_TM_ALLELE_DIFFERENCE=10, then a set of oligos with difference in Tm of 9℃ will be penalized 3 points, a set of oligo s with a difference of 8℃ will be penalized 6 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_TM_ALLELE_DIFFERENCE_SCALE=LINEAR

The scale of OLIGO_TM_ALLELE_DIFFERENCE_SCALE may be linear or square. When the weight is set to 3, a linear scale will give a 6 point penalty to a design where the difference in Tm of a mishybridization between a probe (e.g. wild type) and an allele (e.g. mutant) is 2℃ less than the OLIGO_OPT_TM_ALL ELE_DIFFERENCE. In the example, this is 8℃. A square scale will give a 9 point penalty to the same design.

OLIGO_MIN_DG_ALLELE_DIFFERENCE=0.5

When designing a PROBE PAIR (or ALLELE SPECIFIC PCR or BEACON PAIR) it is necessary to quantify the minimum amount that the two probes/primers (wild type and mutant) must differ from each other. This can be done by Tm or dG. The example shows that a mutant probe/primer must bind to the wild type target with a dG at least 0.5 kcal/mole weaker (more positive) than the hybridization with the mutant target, and vice versa. The OLIGO_MIN_DG_ALLELE_DIFFERENCE is a hard limit and in the example, AOI2Server will not consider oligos whose allelic mishybridizations have a difference in dG less than 0.5 kcal/mole.

OLIGO_OPT_DG_ALLELE_DIFFERENCE=5

OLIGO_OPT_DG_ALLELE_DIFFERENCE is defined as the optimal dG in kcal/mole that one probe (e.g. wild type) can mishybridize against the other target allele (e.g. mutant). The OLIGO_OPT_DG_ALLELE_DIFFERENCE is a soft target; AOI2Server will prefer oligos with allelic mishybridization differences at this dG (in the example, it is 5 kcal/mol) over oligos with other allelic mushybridization dGs.

OLIGO_DG_ALLELE_DIFFERENCE_WT=3

This is the relative penalty (weight) that AOI2Server places on oligos per kcal/mole that the dG difference between 2 probes is less or greater than OLIGO_OPT_DG_ALLELE_DIFFERENCE. Therefore if an OLIGO_OPT_DG_ALLELE_DIFFERENCE=5, then a set of oligos with difference in dG of 4 kcal will be penalized 3 points, a set of oligos with a difference of 3 kcal/mole will be penalized 6 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_DG_ALLELE_DIFFERENCE_SCALE=LINEAR

The scale of OLIGO_DG_ALLELE_DIFFERENCE_SCALE may be linear or square. When the weight is set to 3, a linear scale will give a 6 point penalty to a design where the difference in dG between a probe (e.g. wild type) and an allele (e.g. mutant) is 2 kcal less than the OLIGO_OPT_DG_ALLELE_DIFFERENCE. In the example, this is 3 kcal/mole. A square scale will give a 9 point penalty to the same design.

Parameters to define what significant BLAST hits are

OLIGO_OPT_BLAST_HITS=0

OLIGO_OPT_BLAST_HITS is defined as the optimal number of BLAST hits that an oligo design (probes or primers) is allowed against the database specified by the BLAST_DATABASE keyword. The OLIGO_OPT_BLAST_HITS is a soft target; AOI2Server will prefer oligos with this many BLAST hits over other designs. It is recommended that the default of 0 is maintained unless under specific advanced conditions.

OLIGO_MAX_BLAST_HITS=10

OLIGO_MAX_BLAST is defined as the maximum number of BLAST hits that any oligo (primer or probe) may have against the database specified by the BLAST_DATABASE keyword in order to still be considered a possible solution. OLIGO_MAX_BLAST_HITS is a hard limit and in the example, AOI2Server will not consider oligos that have more than 10 BLAST hits. The valid range is 1 to 9999.

OLIGO_BLAST_HITS_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per BLAST hit that an oligo has more than the OLIGO_OPT_BLAST_HITS. Therefore if an OLIGO_OPT_BLAST_HITS=0, then an oligo with 3 BLAST hits will be penalized 3 points, and an oligo with 5 hits would be penalized 5 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_BLAST_HITS_WT_SCALE=LINEAR

The scale of OLIGO_BLAST_HITS_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where there are 5 more BLAST hits than the OLIGO_OPT_BLAST_HITS. A square scale will give a 25 point penalty to the same design.

OLIGO_OPT_MATCH_LENGTH=0

OLIGO_OPT_BLAST_HITS is defined as the optimal number of matching basepairs for a BLAST hit for an oligo design (probes or primers). OLIGO_OPT_MATCH_LENGTH is a soft target; AOI2Server will prefer oligos with BLAST hits containing this many nucleotide matches before considering other oligos. It is recommended that the default of 0 is maintained unless under specific advanced conditions.

OLIGO_MAX_MATCH_LENGTH=15

OLIGO_MAX_MATCH_LENGTH is defined as the maximum number of matching basepairs that a single BLAST hit for an oligo design (primer or probe) may have to be considered a possible solution. OLIGO_MAX_MATCH_LENGTH is a hard limit and in the example, AOI2Server will not consider an oligo that has a BLAST hit which has a match length of more than 15. It is possible to set the OLIGO_MAX_MATCH_LENGTH to a length based parameter. Therefore OLIGO_MAX_MATCH_LENGTH=LENGTH-5 is a legal statement. If a "LENGTH" parameter is used, then "LENGTH", must be immediately followed by an operand -, /, or *, then a number. (e.g. LENGTH-5 or LENGTH/2, etc.)

OLIGO_MATCH_LENGTH_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per nucleotide that an oligo has a matching basepair more than the OLIGO_OPT_MATCH_LENGTH. Therefore if an OLIGO_OPT_MATCH_LENGTH=0, then an oligo with a single BLAST hit with a match of 8 would be penalized 8 points (based on the example value above). These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_MATCH_LENGTH_WT_SCALE=LINEAR

The scale of OLIGO_MATCH_LENGTH_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to an oligo design which has a single BLAST hit with a match of 5 nucleotides more than OLIGO_OPT_MATCH_LENGTH. A square scale will give a 25 point penalty to the same design.

Parameters to define what significant ThermoBLAST hits are (only active if the ThermoBLAST module is licensed)

OLIGO_OPT_THERMOBLAST_HITS=0

OLIGO_OPT_THERMOBLAST_HITS is defined as the optimal number of ThermoBLAST hits that an oligo design (probes or primers) is allowed against the database specified by the THERMOBLAST_DATABASE keyword. The OLIGO_OPT_THERMOBLAST_HITS is a soft target; AOI2Server will prefer oligos with this many ThermoBLAST hits over other designs. It is recommended that the default of 0 is maintained unless under specific advanced conditions.

OLIGO_MAX_THERMOBLAST_HITS=10

OLIGO_MAX_THERMOBLAST is defined as the maximum number of ThermoBLAST hits that any oligo (primer or probe) may have against the database specified by the THERMOBLAST_DATABASE keyword in order to still be considered a possible solution. OLIGO_MAX_THERMOBLAST_HITS is a hard limit and in the example, AOI2Server will not consider oligos that have more than 10 BLAST hits.

OLIGO_THERMOBLAST_HITS_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on oligos per ThermoBLAST hit that an oligo has more than the OLIGO_OPT_THERMOBLAST_HITS. Therefore if an OLIGO_OPT_THERMOBLAST_HITS=0, then an oligo with 3 ThermoBLAST hits will be penalized 3 points, and an oligo with 5 hits would be penalized 5 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_THERMOBLAST_HITS_WT_SCALE=LINEAR

The scale of OLIGO_THERMOBLAST_HITS_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where there are 5 more ThermoBLAST hits than the OLIGO_OPT_THERMOBLAST_HITS. A square scale will give a 25 point penalty to the same design.

Stem specific parameters for closed beacon designs

OLIGO_STEM_MIN_SIZE=4

OLIGO_STEM_MIN_SIZE is defined as the minimum size in nucleotides of each stem of the beacon. OLIGO_STEM_MIN_SIZE is a hard limit and in the example, AOI2Server will not consider stems less than 4 nucleotides.

OLIGO_STEM_OPT_SIZE=5

OLIGO_STEM_OPT_SIZE is defined as the optimal size in nucleotides of each stem of the beacon. The OLIGO_STEM_OPT_SIZE is a soft target; AOI2Server will favor stems of this length (in the example, it is 5 nucleotides) over stems of other lengths.

OLIGO_STEM_MAX_SIZE=7

OLIGO_STEM_MAX_SIZE is defined as the maximum size in nucleotides of each stem of the beacon. OLIGO_STEM_MAX_SIZE is a hard limit and in the example, AOI2Server will not consider stems greater than 7 nucleotides.

OLIGO_STEM_SIZE_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on stems per nucleotide that a stem is less than the optimal stem length. This weight is calculated for stems that are less than the OLIGO_STEM_OPT_SIZE. Therefore if an OLIGO_STEM_OPT_SIZE=5, then a 4 nucleotide stem will be penalized 1 point, a 3 nucleotide stem will be penalized 2 points, and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_STEM_SIZE_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on stems per nucleotide that a stem is greater than the optimal length. This weight is calculated for stems that are greater than the OLIGO_STEM_OPT_SIZE. Therefore if an OLIGO_STEM_OPT_SIZE=5, then a 6 nucleotide stem will be penalized 1 point, a 7 nucleotide stem will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_STEM_SIZE_WT_SCALE=LINEAR

The scale of OLIGO_STEM_SIZE_WT_SCALE may be linear or square. A linear scale will give a 2 point penalty to a design where the stem of the beacon is 2 nucleotides greater than the OLIGO_STEM_OPT_SIZE. In the example, this is 7 nucleotides. A square scale will give a 4 point penalty to the same design.

OLIGO_STEM_MIN_DG=-10

OLIGO_STEM_MIN_DG is defined as the minimum dG in kcal/mole of each stem of the beacon. OLIGO_STEM_MIN_DG is a hard limit and in the example, AOI2Server will not consider stems with a dG less (more negative) than -10 kcal/mole.

OLIGO_STEM_OPT_DG=-6

OLIGO_STEM_OPT_DG is defined as the optimal dG in kcal/mole of each stem of the beacon. The OLIGO_STEM_OPT_DG is a soft target; AOI2Server will favor stems of this dG (in the example, it is 5 nucleotides) over stems of other dG's.

OLIGO_STEM_MAX_DG=-4

OLIGO_STEM_MAX_DG is defined as the maximum dG in kcal/mole of each stem of the beacon. OLIGO_STEM_MAX_DG is a hard limit and in the example, AOI2Server will not consider stems with a dG greater (more positive) than -4 kcal/mole.

OLIGO_STEM_DG_LT_WT=1

This is the relative penalty (weight) that AOI2Server places on stems per kcal/mole that a stem is less than the optimal stem dG. This weight is calculated for stems that are less than the OLIGO_STEM_OPT_DG. Therefore if an OLIGO_STEM_OPT_DG=-6, then a stem with a dG of -8 will be penalized 2 points, a stem with a dG of -9 will be penalized 3 points, and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_STEM_DG_GT_WT=1

This is the relative penalty (weight) that AOI2Server places on stems per kcal/mole that a stem is greater than the optimal stem dG. This weight is calculated for stems that are greater than the OLIGO_STEM_OPT_DG. Therefore if an OLIGO_STEM_OPT_DG=-6, then a stem with a dG of -5 will be penalized 1 point, a stem with a dG of -4 will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_STEM_DG_WT_SCALE=LINEAR

The scale of OLIGO_STEM_DG_WT_SCALE may be linear or square. A linear scale will give a 2 point penalty to a design where the stem of the beacon is 2 kcal/mole greater than the OLIGO_STEM_OPT_DG. In the example, this is a dG of -4 kcal/mole. A square scale will give a 4 point penalty to the same design.

NUM_REDUNDANT_STEMS=100

The number of candidate stems that are passed from one heuristic to another is based on the NUM_REDUNDANT_STEMS. The NUM_REDUNDANT is a keyword that is necessary to ensure that all stem solutions do or do not contain redundant solutions. Therefore, using default values, the top solutions will differ significantly from other solutions.

OLIGO_STEM_SOLUTION_DISTANCE=1

The distance in nucleotides that one stem can be away from an adjacent solution. If the OLIGO_STEM_SOLUTION_DISTANCE=1, then AOI2Server will look for a stem and then consider the next feasible stem at least 1 nucleotide away (in this example). With this parameter one can ensure that stems will not have high amounts of similarity (by increasing the value).

PROBE_STEM_TERMINAL_GC=FALSE

Indicates if the stem of a beacon should have a closing GC basepair, this is enforced when set to TRUE; both AT and GC closing basepairs are allowed when set to FALSE.

OLIGO_STEM_MIN_GCC=20

OLIGO_STEM_MIN_GCC is defined as the minimum GC content in percentage of each stem. OLIGO_STEM_MIN_GCC is a hard limit and in the example, AOI2Server will not consider stems with a GC content less than 20%.

OLIGO_STEM_OPT_GCC=55

OLIGO_STEM_OPT_GCC is defined as the optimal GC content in percentage of each stem. The OLIGO_STEM_OPT_GCC is a soft target; AOI2Server will prefer stems of this GC content (in the example, it is 55%) over stems with other GC percentages.

OLIGO_STEM_MAX_GCC=80

OLIGO_STEM_MAX_GCC is defined as the maximum GC content in percentage of each stem. OLIGO_STEM_MAX_GCC is a hard limit and in the example, AOI2Server will not consider stems with a GC content greater than 80%.

OLIGO_STEM_GCC_LT_WT=0.1

This is the relative penalty (weight) that AOI2Server places on stems per percent that a stem is less than the optimal stem GC content percentage. This weight is calculated for stems with a GC content less than the OLIGO_STEM_OPT_GCC. Therefore if an OLIGO_STEM_OPT_GCC=55, then a stem with a GC content of 54% will be penalized 1 point, and a stem with a GC content of 53% will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_STEM_GCC_GT_WT=0.1

This is the relative penalty (weight) that AOI2Server places on stems per percent that a stem is greater than the optimal stem GC content percentage. This weight is calculated for stems with a GC content greater than the OLIGO_STEM_OPT_GCC. Therefore if an OLIGO_STEM_OPT_GCC=55, then a stem with a GC content of 56% will be penalized 1 point, and a stem with a GC content of 57% will be penalized 2 points and so on. These values are relative to other weights for other factors (e.g. GC content) and are tabulated in an overall scoring system on which oligos are ranked (best candidate is the one with the fewest penalty points).

OLIGO_STEM_GCC_WT_SCALE=LINEAR

The scale of OLIGO_STEM_GCC_WT_SCALE may be linear or square. A linear scale will give a 5 point penalty to a design where the GC content in % of the stem is 5% greater than the OLIGO_STEM_OPT_GCC. In the example, this is 60%. A square scale will give a 25 point penalty to the same design.

Chapter 3. Output (OOF, TBS, TA, NAL and OSF) Files

3.1 OMP Output Files

OOF (OMP Output File) Output Files

ASSAY_TEMPERATURE=55
SALT_CONCENTRATION=0.05
MAGNESIUM_CONCENTRATION=0.0025

Conditions of the simulation echoed from the Input .oef file.

TOTAL_UNINTENDED_EXTENSIBLE_CONCENTRATION=1.3409e-008

The total concentration of species that are considered extensible, excluding intended heterodimers.

SPECIES=MONOMER

The type of species. The most common values are: monomer, homodimer, heterodimer, trimer and tetramer. The SPECIES keyword appears multiple times in an .oof in general.

SEQUENCE_NAME(S)=Target_1

The name of the complex. In multiple oligo species, individual oligos are delimited by a "+" sign. The SEQUENCE_NAME(S) keyword appears multiple times.

STRUCTURE=OPTIMAL

The structure may be optimal, suboptimal or uncorrected. Optimal is defined as the thermodynamically most likely structure to exist for the SPECIES. Suboptimal structures represent structures that are thermodynamically significant but do not make up as large a percentage of the SPECIES as the optimal. Suboptimal parameters are defined in the Input section in the .oef. If a structure is uncorrected, the thermodynamics that follow this header are intermediate thermodynamics. Except for the most advanced user, uncorrected structure should be ignored.

DELTA-G=-1.2
DELTA-H=-30.6
DELTA-S=-98.6081
MELTING_TEMPERATURE=37.2

The thermodynamic parameters of SEQUENCE_NAME(S) at the solution conditions set in the .oef. dG represents the change in Gibbs Free Energy, dH represents the change in enthalpy, dS represents the change in entropy. There are no maximum or minimum values for the above values. In general, the melting temperature is the temperature at which 50% of the limiting reagent is bound in the designated complex given that there are no intermediate complexes. All of these thermodynamic keywords appear for every species and are likely to appear multiple times in one output file.

CONCENTRATION=2.97939e-010
PERCENT_BOUND=1.19175

Numerical analysis of a SEQUENCE_NAME(S) in a simulation. The CONCENTRATION is defined as the concentration of SEQUENCE_NAME(S) in the SPECIES at the solution conditions set in the .oef. The PERCENT_BOUND can be defined as the percentage of SEQUENCE_NAME(S) that exists in the SPECIES at the solution conditions set in the .oef. If the SPECIES is a monomer, then CONCENTRATION and PERCENT_BOUND refer to the folded monomer. If the SPECIES is a heterodimer, trimer or tetramer, then the PERCENT_BOUND reflects how much of the first listed sequence (in the SEQUENCE_NAME(S)) exists in this SPECIES. These keywords appear for each SEQUENCE_NAME(S).

CONCENTRATION_RANDOMCOIL=3.93114e-011
PERCENT_RANDOMCOIL=0.157245

These keywords only appear if the SPECIES is a monomer. These keywords refer to the concentration and percent bound of a random coiled (secondary structure free) structure. These keywords appear for each monomer.

PERCENT_BOUND(2)=11.917

If the SPECIES is a heterodimer, trimer or tetramer, then the PERCENT_BOUND(2) reflects how much of the second listed sequence (in the SEQUENCE_NAME(S)) exists in this SPECIES.

TARGET_START_POS=50

The start position of the 3' terminus of a probe/primer on a target. Therefore, if a target is 105 nucleotides, and the TARGET_START_POS=50, then the probe/primer has a 3' at the 50th nucleotide of the target.

NET_DG=0.963757

The dG of this SPECIES that results from the multi-state equilibria calculations (numerical analysis) that take competing species into account.

NET_TM=66

The Tm of this SPECIES that results from the multi-state equilibria calculations (numerical analysis) that take competing species into account.

TEMPLATE_STRAND=Target_1

The SEQUENCE_NAME(S) against which an oligo is extensible. Therefore, in the following duplex example, the TEMPLATE_STRAND is Target_1. The TEMPLATE_STRAND is used to create the "extension" of the extensible end. More than one TEMPLATE_STRAND may be declared for heterodimers. Homodimers are by their nature symmetrical and only have one TEMPLATE_STRAND.

```
      Target_1
5' ACTGCATGCGTGCTGA 3'
           3' ACGACT 5'
      Target_2
```

EXTENSION_SITE=10

The extension site on the TEMPLATE_STRAND is the index of the nucleotide from which the extensible end (on Target_2 in the example) is extended. Therefore in the above duplex example, the EXTENSION_SITE is a "G" on Target-1. More than one TEMPLATE_SITE may be declared for heterodimers. Homodimers are by their nature symmetrical and only have one TEMPLATE_SITE.

EXTENSION_OVERHANG=0

The overhang on the 3' end of the extensible site; 0 means a blunt end (e.g. paired nucleotide on the 3' end) and 1 means a terminal mismatch on the 3' end.

INTENDED_TARGET=TRUE

If the species is a heterodimer, the two constituents are a probe or primer and a target, and the INTENDED_TARGET is correctly used in the input file, the heterodimer may be labeled as TRUE. The INTENDED_TARGET=TRUE keyword will identify target-probe heterodimers that are of interest.

INTENDED_POSITION=TRUE

If the species is a heterodimer, the two constituents are a probe or primer and a target, and the INTENDED_POSITION is correctly used in the input file, the position of the probe/primer on the target may be labeled as TRUE. The INTENDED_POSITION keyword will identify target-probe (or primer) heterodimers where the probe/primer is bound at the intended target at the intended position.

NETTM_NAME=CustomNetTM1

The name of the custom NetTM setting.

NETTM_FORMULA=Target_1,Probe_3

The sequences involved in the NetTm. Sequences are delimited by a comma. The concentrations of the species in NETTM_FORMULA are aggregated when determining the NET_MELTING_TEMPERATURE.

NETTM_LIMITING_SEQUENCE=Target_1

The sequence from which the NetTm was calculated. In this example, the NET_MELTING_TEMPERATURE was determined when 40% (NETTM_THRESHOLD) of Target_1 was involved in the species described by NETTM_FORMULA.

NETTM_THRESHOLD=40

The percentage of NETTM_LIMITING_SEQUENCE at which the NET_MELTING_TEMPERATURE is calculated. In this example, the NET_MELTING_TEMPERATURE Is calculated when 40% of the NETTM_LIMITING_SEQUENCE is bound in the species defined by the NETTM_FORMULA.

NET_MELTING_TEMPERATURE=75.2

The temperature at which the species in the NETTM_FORMULA contain the NETTM_THRESHOLD of the NETTM_LIMITING_SEQUENCE. Using the above parameters as an example, the NET_MELTING_TEMPERATURE is the temperature at which 40% of Target_1 is in the Target_1+Probe_3 duplex. If the NETTM_THRESHOLD is not met, then a "-1" is returned. This means that the NET_MELTING_TEMPERATURE does not exist.

TBS (Traceback Structure) Output Files

```
ASSAY_TEMPERATURE=55
SODIUM_CONCENTRATION=0.05
MAGNESIUM_CONCENTRATION=0.0015
GLYCEROL_CONCENTRATION=0
DMSO_CONCENTRATION=0
FORMAMIDE_CONCENTRATION=0
TMAC_CONCENTRATION=0
BETAINE_CONCENTRATION=0
```

Conditions of the simulation echoed from the Input .oef file.

```
SEQUENCE_NAME(S)=TARGET+PROBE
```

The name of the complex. In multiple oligo species, individual oligos are delimited by a "+" sign

```
SEQUENCE(S)=CTAGCAGCTAGCGATGCTAGCTAGCTAGCTAGAGCTAGCTGACTGACTGCATLLLAGTCAGTCAGCTAGCTCTAGC
```

The sequence for which the basepairs will be listed below. The sequence will be combined from individual sequences in multiple oligo species; here the sequences of TARGET and PROBE are combined.

```
SEQUENCE_NAME=TARGET
CONCENTRATION=1e-007
SEQUENCE_NAME=PROBE
CONCENTRATION=1e-006
```

Name and concentration of individual oligos that make up SEQUENCE_NAME(S).

```
NUMBER_STRUCTURES=4
```

The number of structures for which the basepairs are listed further below for this SEQUENCE_NAME(S) species.

```
STRUCTURE=OPTIMAL
```

Can be OPTIMAL, SUBOPTIMAL#1, etc. This indicates to which structure of the species the basepairs following below refer to.

```
BASEPAIR COUNT =28
```

The number of basepairs formed for the SEQUENCE_NAME(S) species.

```
BASEPAIRS#1=28 , 76
BASEPAIRS#2=29 , 75
```

Lists the nucleotide numbers of both bases a basepair consists of for all basepairs in the structure.

```
DELTA-G=-16.489
DELTA-H=-228.2
DELTA-S=-645.165
MELTING_TEMPERATURE=66.42
```

The thermodynamic parameters of SEQUENCE_NAME(S) at the solution conditions set in the .oef. dG represents the change in Gibbs Free Energy, dH represents the change in enthalpy, dS represents the change in entropy.

TA (Target Accessibility/Complexity) Output Files

TA-files are only generated for sequences of SEQUENCE_FUNCTION=TARGET and contain information on the accessibility and complexity of the target for windows of a certain nucleotide size. It is a tab delimited file with the following contents:
Line 1 lists the column heading and general information. The first 3 columns will then contain the accessibility and complexity information for the individual windows from line 2 on.

Line 1 headings and information:

- StartPos: number of the window for which the accessibility and complexity are calculated
- ThermoAccess: energy needed to unfold this part of the target, the lower the less secondary structure, and the more accessible the target is at this location
- Complexity: base variety in the window for which the accessibility and complexity are calculated
- 20= ThermoAccess Window Size: size (in nucleotides) of the windows used to calculate Accessibility
- 20= Complexity Window Size: size (in nucleotides) of the windows used to calculate complexity
- OLIGO1: name of the Target sequence for which Accessibility and Complexity are listed in this TA-file

From Line 2 on, the first 3 columns will contain the values for subsequent windows for both Accessibility (column 2) and Complexity (column 3).

NAL (Numerical Analysis) Output Files

NAL-files list the concentrations for all possible species for a range of temperatures in a tab-delimited format.

NUMANALY_MAX_TEMPERATURE=100
NUMANALY_MIN_TEMPERATURE=10

Repeats the settings for the temperature range from the input .oef file.

In the next section, TEMPERATURE indicates at which temperature the concentrations were calculated for this row. It starts at the NUMANALY_MAX_TEMPERATURE and goes down by 1 degree decrements to the NUMANALY_MIN_TEMPERATURE. The other columns list the concentrations for the species present, their SEQUENCE_NAME(S) are listed at the top of the columns.

3.2 AOI2Server Design Output (OSF)

AOI2Server Design output files (OSFs) contain a relatively straightforward architecture. The first half of the output file contains all of the parameters from the input file (ODF) used to determine the probes and/or primers. The keywords and explanations for each can be found in chapter 2.2 AOI2Server Design. Directly following the parameters is a section that summarizes the design experiment. Lastly, the solutions to the design are listed. This documentation only shows the output for a single solution; however, most outputs will likely contain more than one solution. If a multiplex design is performed, two oligos with different types (e.g. primer pair and probe) may have the same solution number.

3.2.1 Parameters

Please see chapter 2.2 AOI2Server Design for explanations of keywords.

3.2.2 Design Experiment Summary

OLIGO_NAME=OLIGO_1

The name of the oligo (probe or primer).

OLIGO_TARGET_NAME=ATP7B

The name of the target against which the oligo (probe or primer) was designed.

OLIGO_TARGET_STRAND=SENSE

The target strand against which the oligo (probe or primer) was designed.

OLIGO_TYPE=PROBE

The nature of the designed oligo. The OLIGO_TYPE may be: PROBE, PROBE PAIR, PRIMER PAIR, FORWARD PRIMER, REVERSE PRIMER, ALLELE SPECIFIC PCR, FIXED PROBE, TAQMANPROBE, TAQMAN_MGB_PROBE, BEACON, or BEACON PAIR.

OLIGO_PARAMETER_PROFILE=Balanced

The strategy employed by Visual OMP to set the parameters for design. As mentioned in section chapter 2.2.3, the OLIGO_PARAMETER_PROFILE keyword is used by Visual OMP and for the purposes of AOI2Server only used as a comment.

OLIGO_CANDIDATES=484

The number of possible oligos (probes or primers) that were considered for the design experiment. This number is based mostly on length min/max values.

EXCLUDED_BY_SOLUTION_DISTANCE_FILTER=237

The number of candidate oligos not considered due to the SOLUTION_DISTANCE keyword in the .odf.

OLIGO_HAIRPIN_FAILURES=11

The number of candidates that failed hairpin checks.

OLIGO_HOMODIMER_FAILURES=0

The number of candidates that failed homodimer checks.

OLIGO_LOCAL_TARGET_FAILURES=10

The number of candidates that failed because of local secondary structure on the target.

OLIGO_MISHYB_FAILURES=0

The number of candidates that failed due to mishybridization.

OLIGO_MONOMER_EXTENSION_FAILURES=4

The number of candidates that failed due to unwanted extensible monomer structures.

OLIGO_HOMODIMER_EXTENSION_FAILURES=3

The number of candidates that failed due to unwanted extensible homodimer structures.

OLIGO_HETERODIMER_EXTENSION_FAILURES=1

The number of candidates that failed due to unwanted extensible heterodimer structures.

3.2.3 Design Solutions

SOLUTION_NUMBER=1

The number of the solution. If the OEF requested 5 solutions, then each solution will be listed in the OSF as SOLUTION_NUMER=1, SOLUTION_NUMBER=2 and so on.

SCORE=987.279

The overall score of the complete solution. The closer to 1,000, the fewer penalty points were assigned during design, and the better the optimal design parameters have been met. The first solution always has the highest Q-score.

OLIGO_SOLUTION_NUMBER=1

The number of the solution, repeated for each oligo within one solution.

OLIGO_SCORE=987.279

The score of the individual oligo. The closer to 1,000, the fewer penalty points were assigned to it during design.

OLIGO_NAME=OLIGO_1

The name of the designed oligo.

OLIGO_TYPE=PROBE

The nature of the designed oligo. The OLIGO_TYPE may be: PROBE, PROBE PAIR, PRIMER PAIR, FORWARD PRIMER, REVERSE PRIMER, ALLELE SPECIFIC PCR, FIXED PROBE, TAQMANPROBE, TAQMAN_MGB_PROBE, BEACON, or BEACON PAIR.

OLIGO_SEQUENCE_TYPE=DNA

The backbone type of the designed oligo, usually DNA or RNA. PNA, Mor, PSD, and OM2 are also possible if the Modifieds module is licensed.

OLIGO_TARGET_NAME=ATP7B

The name of the target against which the oligo (probe or primer) was designed.

OLIGO_TARGET_STRAND=SENSE

The target strand against which the oligo (probe or primer) was designed.

OLIGO_SEQUENCE=AGCGTGGTGTTAAAGT

The sequence of the oligo (probe or primer)

TARGET_START_POS=77

The start position of the oligo's (probe or primer) 3' end on the target. The start nucleotide in this example is the 77th nucleotide from the 5' terminus of the sense strand of the target.

TARGET_ANTISENSE_START_POS=5

The start position of the oligo's (probe or primer) 5' end on the target. The start nucleotide in this example is the 5th nucleotide from the 5' terminus of the antisense strand of the target.

OLIGO_LENGTH=16

The length of the oligo (probe or primer).

OLIGO_TM=72.4809
The melting temperature of the oligo bound to the target considering only the target and the oligo (duplex Tm).

OLIGO_DG=-11.942
The dG of the oligo bound to the target considering only the target and the oligo (duplex dG).

OLIGO_GCC=43.75
The GC content of the oligo.

OLIGO_MONOMER_TM=8.31384
The melting temperature of the folded oligo(probe or primer).

OLIGO_MONOMER_DG=2.461
The dG of the folded oligo (probe or primer).

OLIGO_TARGET_LOCAL_TM=-4.06244
The Tm of the local folded target segment. The length of the local segment is defined in the OEF.

OLIGO_TARGET_LOCAL_DG=1.295
The dG of the local folded target segment. The length of the local segment is defined in the OEF.

MISHYB_TEST=PASS
Indicates if the mishybridization criteria were satisfied by this oligo design.

HAIRPIN_TEST=PASS
Indicates if the hairpin/monomer criteria were satisfied by this oligo design.

DIMER_TEST=PASS
Indicates if the homodimer/self dimer criteria were satisfied by this oligo design.

LOCALTARGET_HAIRPIN_TEST=PASS
Indicates if the local target criteria were satisfied by this oligo design.

COMPUTATION_TIME=2.734 s
The time taken to run the design experiment.