



**Doelan v.1.0.1**

**Project Documentation**



# Table of Contents

---

## 1 About Doelan

1.1 What is Doelan ?	1
1.2 Screenshots	2
1.3 License	3
1.4 People behind Doelan	9

## 2 Getting Doelan

2.1 Downloading	10
2.2 Installing	11
2.3 Release Notes	13

## 3 Documentation

3.1 QuickStart tutorial	18
3.2 Available tests	24
3.2.1 DoelanConfigure	26
3.2.2 BadFeature	27
3.2.3 AbsentFeature	28
3.2.4 NotFoundFeature	29
3.2.5 MaxBadFeatures	30
3.2.6 MaxBadFeaturesRemoved	31
3.2.7 HeterogeneousFeature	32
3.2.8 SaturatedPixels	33
3.2.9 MinimalIntensity	34
3.2.10 MaxDiameterFeature	35
3.2.11 MinDiameterFeature	36
3.2.12 Generic	37

3.3 Writing your own test .....	38
3.4 PDF documentation .....	43
4 <b>Development</b>	
4.1 Get the source .....	44
4.2 Compiling .....	45
4.3 Known bugs .....	46
4.4 Futur improvements .....	47
4.5 Contribute to Doelan .....	48
5 <b>Community</b>	
5.1 Additional TestSuites .....	49
5.2 Additional Test Plug-ins .....	50
5.3 Mailing list .....	51

## 1.1 What is Doelan ?

---

### Welcome to Doelan

Get Doelan 1.0.1

 [Download](#) Doelan 1.0.1

- [System Requirements](#)
- [Installation Instructions](#)
- [QuickStart tutorial](#)

Doelan is an automated tool to check the quality of produced DNA microarrays. This software is based on the execution of TestSuites on quality control data to validate batches of chips. The reports generated by Doelan will help microarray platforms in their quality approach such as ISO 9001 certification. Doelan application is written with Java programming language, works with a plug-in system that allows everyone to add its own validation test and is distributed under the [GNU General Public License](#).

Production of glass slide microarrays requires the spotting of a large number of probes. In this process, many factors can influence the success of the spotting (i.e. blocked spotting pin, glass slides...); so controlling the quality of the spotting process is a required step before to distribute chips to the user. Up to now, it was quite difficult for the quality control manager to check if the spotting of chips was good or bad: many different parameters as spot diameters, heterogeneous spots or absent spots had to be manually considered to reach a decision. In addition, it is a very subjective step, so that the opinion about a batch may differ between two quality control managers. Using manually defined criterias for microarray quality, Doelan allows an automated expertise of the quality of a batch of slides and makes automatically the decision of validation or rejection of the batch.

The principle of Doelan is based on TestSuites. A TestSuite is a set of tests, which are dedicated to a specific features (for example spot diameter). The microarray designer creates the TestSuites and sets the parameters (i.e. minimal and maximal values, thresholds...) of each test to validate a chip. Several TestSuites can be created according to the type of validation experiment chosen as sybergreen, universal RNA hybridisation or a "real" experiment. The fail of one of the tests leads to the fail of the TestSuite, and consequently to the rejection of all the batch of chips. Doelan is bundled with many tests. These tests can deal with various features of the chips such as spot diameter, spot flags, spot heterogeneity, saturated pixels of the spots... In addition to this standard tests, the Doelan plug-in system with Java programming allows one to set up new kind of test to fulfil specific needs.

## 1.2 Screenshots

---

### Screenshots

This is some screenshots of Doelan on several operating systems:

## 1.3 License

---

### Project License

GNU GENERAL PUBLIC LICENSE  
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE  
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but



does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program

except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates

the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

#### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or modify
```

it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

## 1.4 People behind Doelan

---

### People behind Doelan

Doelan is developed by the bioinformatic team of the microarray platform at the École Normale Supérieure:

Name	Work
Claude Jacq	Transcriptome platform head responsible
Stéphane Le Crom	Bioinformatic responsible and Doelan testing
Laurent Jourden	Doelan design and coding

Special thanks to:

Name	Work
Corinne Blugeon, Fanny Couplier, Frédéric Devaux and Véronique Tanty	For Doelan testing
Charles Hébert	For the new logo

### Contact

Who can you contact if you have any question:

You can also visit our microarray platform [website](#) .

## 2.1 Downloading

---

### Download Doelan latest version

Doelan is distributed in several formats for your convenience.

- [.tar.gz archive](#) (5.4 MiB)
- [.zip archive](#) (5.9 MiB)

Doelan is distributed under the [General Public License](#) .

### System Requirements

Component	Comment
Java Runtime Environment	1.4 or above (Dolean might run under Java 5.0 but no test has been realized).
Memory	No minimum requirement. The necessary amount of memory needed by Doelan depends on size of the files to be tested. Currently with GPR and GAL files greater than 20,000 lines, Doelan doesn't need to modify the memory allocation pool size (default is 64 MiB) with the <b>-Xmx</b> option of the JVM.
Disk Space	No minimum requirement. Approximately 11 MiB will be used by Doelan.
Operating System	No minimum requirement. Tested on Windows XP, Debian testing and Mac OS X.

## 2.2 Installing

---

### Standalone installation

The installation of Doelan on all platform is quite similar and very easy. You don't need to be the administrator or the root user of your system to install Doelan.

The only tool you need to have is a zip extractor. A zip extract tool is included on Windows XP, Linux and Mac OS.

If you are a user of Windows prior to XP, you can download **7-Zip** (7-Zip is a free software) at the following location <http://www.7-zip.org/> .

- [Download Doelan](#) on our website
- Unzip the archive where you want install the software
- Doelan is ready to run.

### Installation for Genepix integration

To configure Genepix integration of Genepix, you need to allow the Doelan applet in Genepix to read and write on your filesystem and set the path to the doelan report page in Genepix.

#### Modify Doelan Applet security permissions

To allow a normal execution of Doelan in Genepix, you need to modify the security permissions for the applet. Edit the `{JAVA_HOME}/lib/security/java.policy` and add the following lines:

```
// Doelan permissions
grant codebase "{PATH_TO_DOELAN}/lib/*" {
    permission java.lang.RuntimePermission "*";
    permission java.io.FilePermission "<<ALL FILES>>", "read";
    permission java.io.FilePermission "<<ALL FILES>>", "write";
    permission java.util.PropertyPermission "*", "read";
};
```

`{JAVA_HOME}` is the path to your java installation and `{PATH_TO_DOELAN}` is the path to Doelan on your filesystem.

#### Add the link to the doelan report page in Genepix

To have an access to Doelan in the report tab of Genepix, you must modify the index of the genepix report file: C:\Axon\Scripts\GenePixPro5\Default.htm

and add the following line in **Analysis Reports** section:


```
<li>
  <a href="{PATH_TO_DOELAN}\genepix-applet.html">
    Doelan Quality Report</a>.
</li>
```









## 2.3 Release Notes

### Release History


Version	Date	Description
<a href="#">1.0.1</a>	2005-09-19	
<a href="#">1.0</a>	2005-07-18	
<a href="#">1.0-rc9</a>	2005-04-15	
<a href="#">1.0-rc8</a>	2005-04-15	
<a href="#">1.0-rc7</a>	2005-04-11	
<a href="#">1.0-rc6</a>		

Get the RSS feed of the last changes 












### Release 1.0.1 - 2005-09-19

Type	Changes	By
	Add information about empty spot in the report.	<a href="#">jourdren</a>
	Fix bug with empty spot in the arrayplot.	<a href="#">jourdren</a>
	Fix bug with scrollbar in html documentation widget.	<a href="#">jourdren</a>
	An array description can be added in the report.	<a href="#">jourdren</a>
	Fix bug with custom empty identifiers.	<a href="#">jourdren</a>
	In minimal intensity test, the channel to test can be set.	<a href="#">jourdren</a>








### Release 1.0 - 2005-07-18

Type	Changes	By
	Fix bug with logarithmic axis.	<a href="#">jourdren</a>





## Release 1.0-rc9 - 2005-04-15

Type	Changes	By
	Use a Pluggable Look And Feel from JGoodies looks under *nix.	<a href="#">jourdren</a>
	Add information pages (about test and documentation of a test) in contextual menu.	<a href="#">jourdren</a>
	New test: GenericTest, generic test for a column of GPR.	<a href="#">jourdren</a>
	New test: GlobalMaxBadFeaturesRemovedTest, count the maximal bad spots removed in a TestSuite	<a href="#">jourdren</a>
	In the output gal file, now, don't remove line but change the the identifier of the line and add in the description field the reason of the reject of the spot.	<a href="#">jourdren</a>
	The arrayplot now show in white, feature which fail to one or more tests but doesn't be removed.	<a href="#">jourdren</a>
	Reorganize the file hierachy of the application. Ready for online update of doelan when available.	<a href="#">jourdren</a>
	User can set the identifiers of the empty spots.	<a href="#">jourdren</a>
	Test identifiers are now in the same order in the TestSuite tab and this the report.	<a href="#">jourdren</a>
	Bad result for spot heterogenity.	<a href="#">jourdren</a>
	Now use JFreechart to draw plot.	<a href="#">jourdren</a>



## Release 1.0-rc8 - 2005-04-15



















Type	Changes	By
	Fix bug for saving and printing report.	<a href="#">jourdren</a>
	Add a framework for global tests.	<a href="#">jourdren</a>
	New Global test: Count all the bad spots in a TestSuite.	<a href="#">jourdren</a>
	Report tab: Show the manufactuer, the name and the revision of the array (informations from GAL file).	<a href="#">jourdren</a>
	Report tab: Add a line in summary section of the report which show the final result of the TestSuite. :-)	<a href="#">jourdren</a>
	Use now romde 0.6-CVS.	<a href="#">jourdren</a>
	Majors changes in romde workflow organization. Now an algorithm before generating the report runs all the globals. Add useful methods to get romde data object from Romde platform container.	<a href="#">jourdren</a>







## Release 1.0-rc7 - 2005-04-11

Type	Changes	By
	Fix huge memory usage by BioAssay Objects in Romde.	<a href="#">jourdren</a>
	ArrayPlot use margin with.	<a href="#">jourdren</a>
	Update maven configuration files.	<a href="#">jourdren</a>
	Fix Graphics plots.	<a href="#">jourdren</a>

## Release 1.0-rc6 -

Type	Changes	By
	Initial realease in CVS.	<a href="#">jourdren</a>
	New Graph : Array visualisation.	<a href="#">jourdren</a>

Type	Changes	By
	Safari problem.	<a href="#">jourdren</a>
	Windows sample plugin compilation script now works.	<a href="#">jourdren</a>
	Make a sample plugin : windows batch.	<a href="#">jourdren</a>
	Make a logo and an icon.	<a href="#">jourdren</a>
	Modifier le script d'integration à Genepix pour qu'il affiche un message d'erreur quand il n'est pas executé dans Genepix.	<a href="#">jourdren</a>
	Change the name of the launcher.	<a href="#">jourdren</a>
	New Graph : Distribution graph.	<a href="#">jourdren</a>
	New test: Spot size.	<a href="#">jourdren</a>
	New test: Empty spot.	<a href="#">jourdren</a>
	New test: Heterogenous spot.	<a href="#">jourdren</a>
	New result output : HTML report.	<a href="#">jourdren</a>
	GUI : Save result array list button.	<a href="#">jourdren</a>
	GUI : Save report button.	<a href="#">jourdren</a>
	New array list output : GAL File.	<a href="#">jourdren</a>
	Change the background of the graph : lighter color	<a href="#">jourdren</a>
	Show the TestSuite name in TestSuite tab and report tab	<a href="#">jourdren</a>
	Show the units of the thresholds.	<a href="#">jourdren</a>
	Show the threshold on the graphs.	<a href="#">jourdren</a>

Type	Changes	By
	New test: Valid spotting.	<a href="#">jourdren</a>
	Polish Report tab	<a href="#">jourdren</a>
	GUI : Print report button.	<a href="#">jourdren</a>
	Clear TestSuite and result tab when start a workflow.	<a href="#">jourdren</a>
	Make a sample plugin (jar + source in docs).	<a href="#">jourdren</a>
	Add a progress bar.	<a href="#">jourdren</a>

## 3.1 QuickStart tutorial

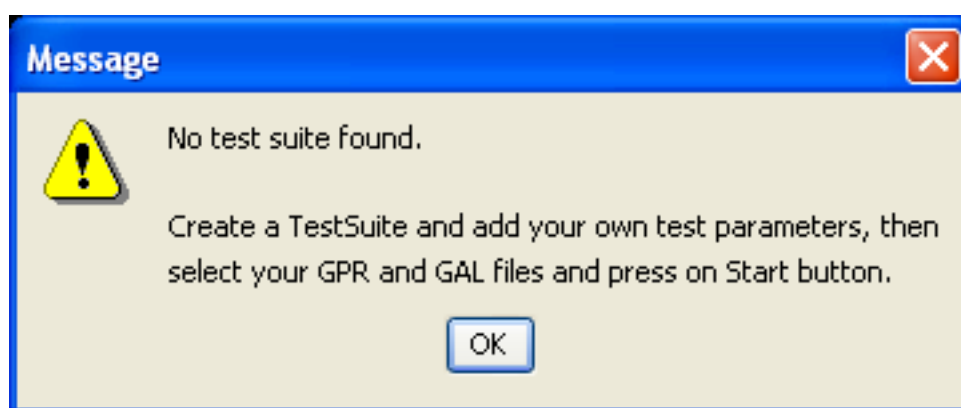
---

### What is a TestSuite?

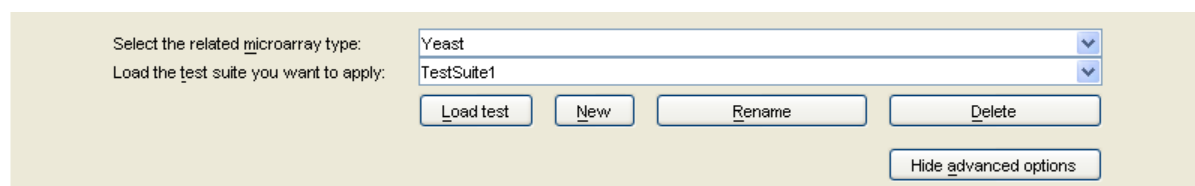
A TestSuite is a collection of tests which are apply on chip to validate it.

### How to create a TestSuite

When you launch Doelan for the first time, a message notify you that you need to first create a TestSuite.



Each TestSuite is link to a chip type, so you need to create a chip type before creating a TestSuite. To create a chip type or TestSuite: on the main tab click on the **New** button, if this button is hidden, you can show this button and other optionals buttons by pressing on the **Show advanced options**. Now a dialog box appears, select the type element which you want to create (Chip type or TestSuite), set a name and validate with **OK** button. If there is no chip type created, you cannot create a TestSuite, you must create a chip type before a TestSuite.



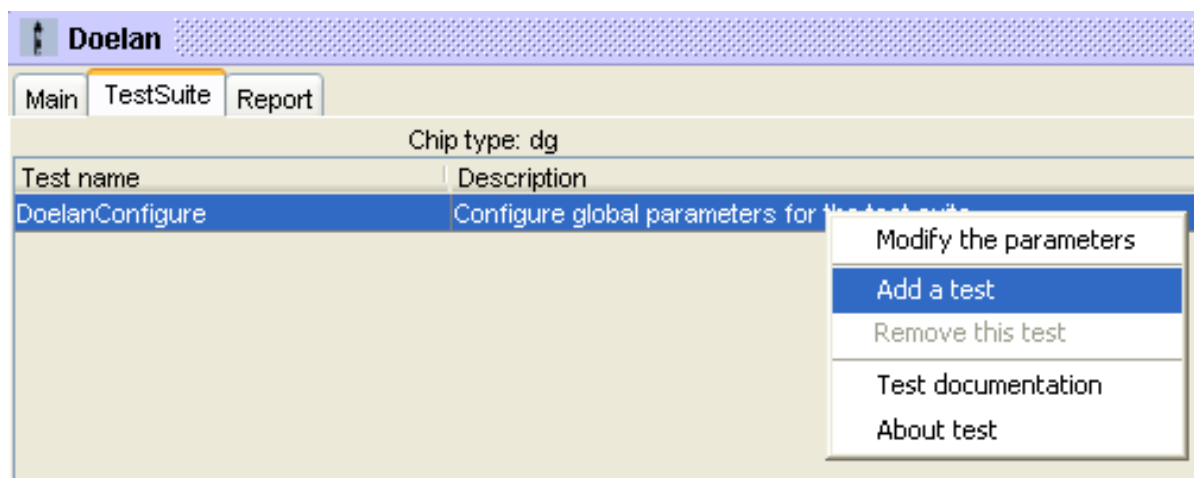
Of course with doelan, in addition to create chip type and TestSuite, you can rename and remove this elements.

### Edit a TestSuite

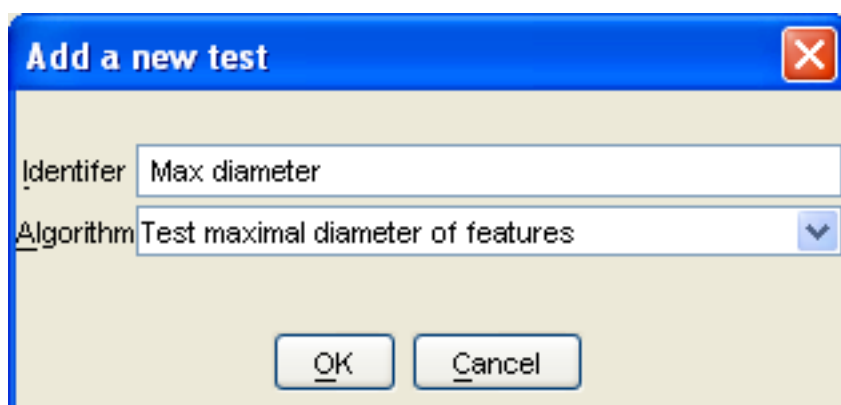
To edit a TestSuite, you must load it first. Loading a TestSuite is performed by a click on the **Load** button or by a click on **Start** button.

If you want to show and modify the elements of a TestSuite, you must go to the **TestSuite** tab of the

application. In a recently created Testuite only one element compose the TestSuite. This element is not really a TestSuite but an element to set global parameters of the TestSuite.

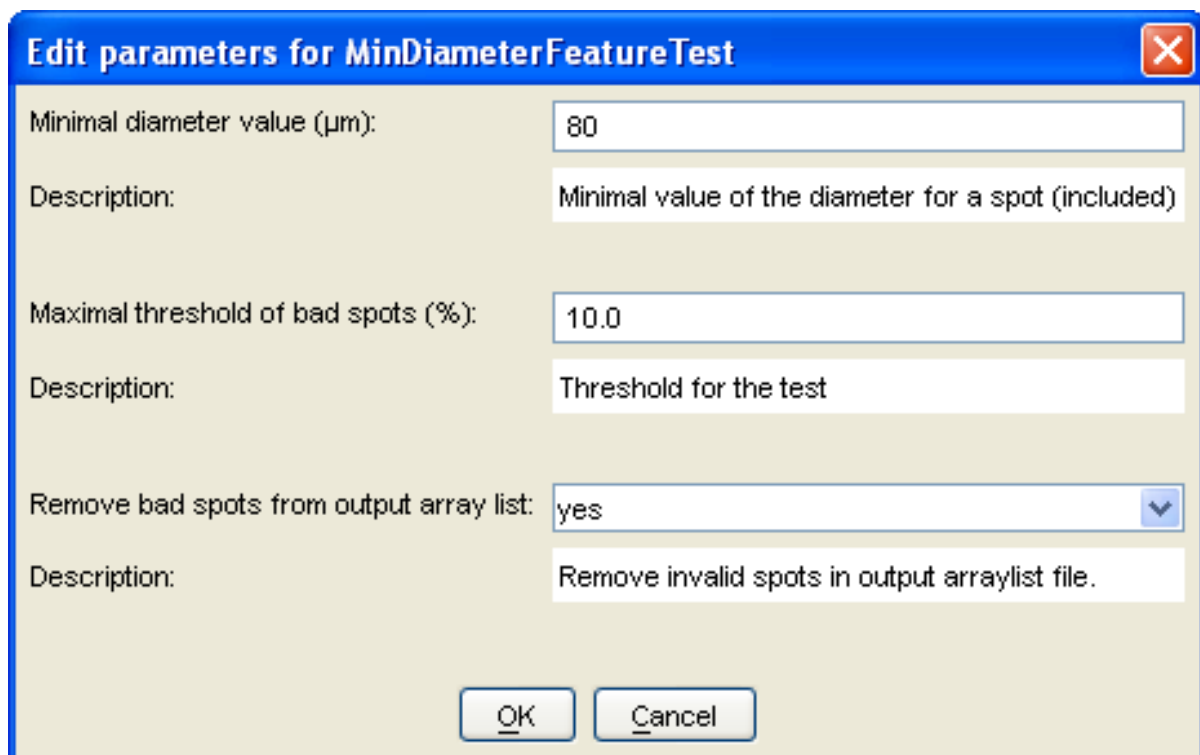


To add a new element, right click (or double click) on an existing element, select **add** and a dialog box appears with the list of all available tests. Select one and define an identifier for this test in your TestSuite. This test identifier must be unique.



There are two kinds of tests in Doelan: global and unit tests. The unit tests provides statistics about chip data while global test provides informations about a set of unit tests results. You can only add one type of global test in a TestSuite.

In a TestSuite you can customize all the parameters of the tests. To do so, right click (or double click) on the test which you want to set the parameters and select **Modify the parameters**.



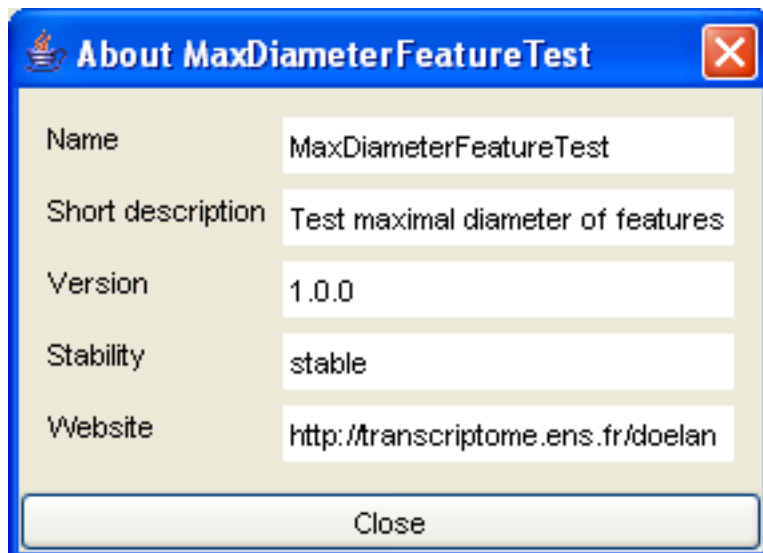
The dialog box has a blue title bar with the text "Edit parameters for MinDiameterFeatureTest" and a close button (X) on the right. The main area is light beige and contains three parameter groups, each with a label, a description, and a value field:

Parameter	Description	Value
Minimal diameter value (µm):	Minimal value of the diameter for a spot (included)	80
Maximal threshold of bad spots (%):	Threshold for the test	10.0
Remove bad spots from output array list:	Remove invalid spots in output arraylist file.	yes

At the bottom, there are two buttons: "OK" and "Cancel".

Go to the [available tests page](#) for more information about test parameters. You can also, of course remove a test in a TestSuite with the **Remove test** option in the contextual menu.

You can show additional information about the test with the **About test** and **Test documentation**.

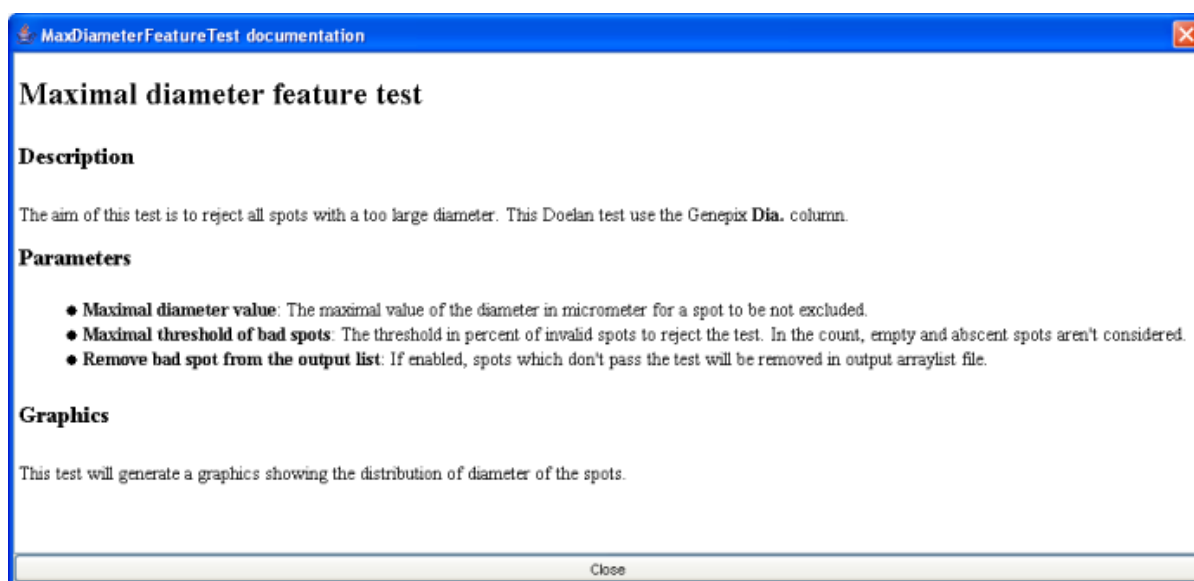


The dialog box has a blue title bar with the text "About MaxDiameterFeatureTest" and a close button (X) on the right. The main area is light beige and contains a table with test details:

Field	Value
Name	MaxDiameterFeatureTest
Short description	Test maximal diameter of features
Version	1.0.0
Stability	stable
Website	<a href="http://transcriptome.ens.fr/doelan">http://transcriptome.ens.fr/doelan</a>

At the bottom, there is a "Close" button.



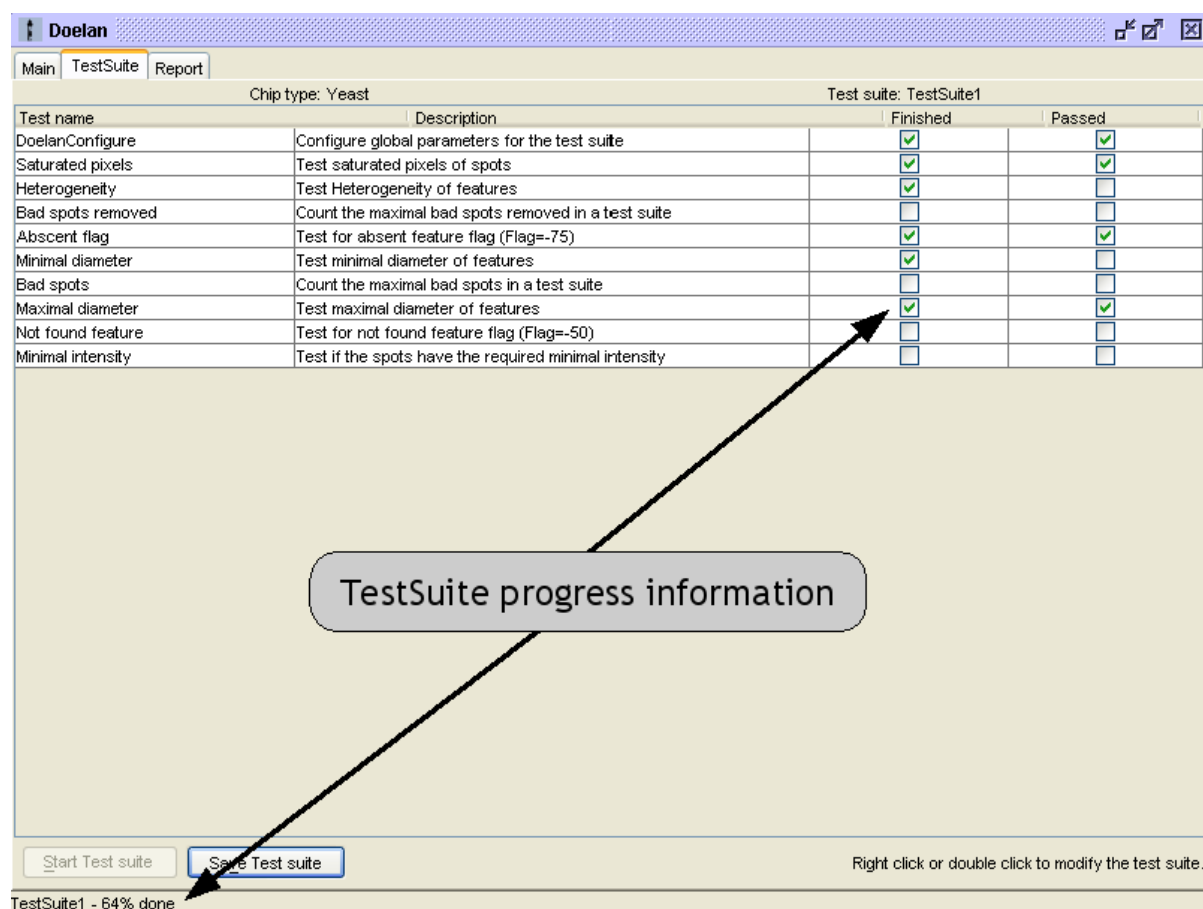


## Launching a TestSuite

To start a TestSuite: in the main tab, select a GPR file and a GAL file, add a comment about the array (optional), select a TestSuite and click on the **Start** button.

Select the GenePix file to analyzed:	Demo.gpr	<input type="button" value="Choose GPR file"/>
Load the Gene Array List (optional):	Demo.gal	<input type="button" value="Choose GAL file"/>
Array description (optional):	<input type="text" value="Demonstration array"/>	

In the status bar and in the TestSuite tab, you can follow the progress of the executed TestSuite. At the end of the execution of a TestSuite a detailed report is available in the report tab.



Giving an array list file (GAL file) is optional but there are two benefits to do so: First, in the report tab you can see an image which represents the locations on the microarray of spots which pass the tests. And secondly, you can save an output array list without rejected spots.

## Report

After launching a TestSuite, Doelan generates a four parts report. The first part shows general informations about the TestSuite used such as the current date or the name of the TestSuite. The second part summaries all the results of the TestSuite. The next part shows detailed informations about all tests with parameters values, the result, a comment and sometimes one or more graph to visualize the results. If an array list file was selected, the last part of the report and represents the locations on the microarray of spots which pass the tests (in green) and which does not (in red).

Doelan

Main TestSuite **Report**

Test name	Description	635	532	Threshold	Result 635	Result 532	Global Result
<a href="#">Saturated pixels</a>	Test saturated pixels of spots	0.00 %	0.00 %	<= 10.00 %	Pass	Pass	Pass
<a href="#">Heterogeneity</a>	Test Heterogeneity of features	26.85 %	28.79 %	<= 10.00 %	Fail	Fail	Fail
<a href="#">Absent flag</a>	Test for absent feature flag (Flag=-75)	0.00 %		<= 5.00 %			Pass
<a href="#">Minimal diameter</a>	Test minimal diameter of features	18.94 %		<= 10.00 %			Fail
<a href="#">Maximal diameter</a>	Test maximal diameter of features	0.29 %		<= 10.00 %			Pass
<a href="#">Not found feature</a>	Test for not found feature flag (Flag=-50)	1.02 %		<= 7.00 %			Pass
<a href="#">Minimal intensity</a>	Test if the spots have the required minimal intensity	0.06 %		<= 10.00 %			Pass
<a href="#">Bad spots</a>	Count the maximal bad spots in a test suite	47.89 %		<= 10.00 %			Fail
<a href="#">Bad spots removed</a>	Count the maximal bad spots removed in a test suite	47.89 %		<= 10.00 %			Fail

Save report Print report Save new Array List

TestSuite1 - done

## Save results

The report can be saved in HTML format or can be printed. If an array list data was selected, the creation of a new array list file without all the rejected spots is allowed. The identifiers of all this spots will be set to a new "empty" identifier and the description field will notify the origin of the rejection. Warning, if this new identifier is in the list of "empty" identifiers, rejected spots in the repport map will be considered as absent (dotted circle) and not as rejected (red filled circle).

Save report	Print report	Save new Array List
-------------	--------------	---------------------

## 3.2 Available tests

---

### Test in Doelan

Doelan is bundled with many tests. However, you can add your own test to Doelan thanks to the Doelan plugin system. The [writing a plugin page](#) learns you how to do this.

### Built-in Test in Doelan

The Doelan builtins tests can play with various features of the chips such spots diameter, spots flags, spots heterogeneity, saturated pixels of the spots...

Here is the list of the built-in tests, click on a test to display documentation on parameters:

- **Unit tests**

Test name	Description
<a href="#">DoelanConfigure</a>	Configure global parameters for the TestSuite
<a href="#">BadFeature</a>	Test for bad feature flag (Genepix flag=-100)
<a href="#">AbsentFeature</a>	Test for absent feature flag (Genepix flag=-75)
<a href="#">NotFoundFeature</a>	Test for not found feature flag (Genepix flag=-50)
<a href="#">HeterogeneousFeature</a>	Test for heterogeneous spots
<a href="#">SaturatedPixels</a>	Test for saturated spots
<a href="#">MinimalIntensity</a>	Test for minimal spot intensity
<a href="#">MaxDiameterFeature</a>	Test maximal diameter of spots
<a href="#">MinDiameterFeature</a>	Test minimal diameter of spots
<a href="#">GenericTest</a>	Generic test to test any column of a GPR file

- **Global tests**

Test name	Description
<a href="#">GlobalMaxBadFeatures</a>	Global test to count rejected spots
<a href="#">GlobalMaxBadFeaturesRemoved</a>	Global test to count removed spots from output file

- **Special**

**Test name**[DoelanConfigure](#)**Description**

Configure global parameters for the TestSuite

## 3.2.1 DoelanConfigure

---

### DoelanConfigure Element

#### Description

This element is not a test but an element used to set several global parameters for the TestSuite.

#### Parameters

- The list of the "empty" identifiers. That is to say, all the identifiers of the spot that Doelan may consider as empty spots.
- Column for empty identifiers. This can be the identifier or the description field.
- The new identifier for the spots which are rejected in the output file. Warning, if this identifier is "empty", rejected spots in the report map will be considered as absent (dotted circle) and not as rejected (red filled circle).

## 3.2.2 BadFeature

---

### Bad feature test

#### Description

The aim of this test is to count and reject all spots with a flag set at -100. This Doelan test use the **Flag** column of the GPR.

#### Parameters

- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

### 3.2.3 AbsentFeature

---

#### Absent feature test

##### Description

The aim of this test is to count and reject all spots with a flag set at -75. This Doelan test use the **Flag** column of the GPR.

##### Parameters

- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.



## 3.2.4 NotFoundFeature

---

### Not found feature test

#### Description

The aim of this test is to count and reject all spots with a flag set at -50. This Doelan test use the **Flag** column of the GPR.

#### Parameters

- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

## 3.2.5 MaxBadFeatures

---

### Global maximum bad features test

#### Description

The aim of this global test is to count the number spots which are rejected by all the unit tests. In this test all rejected spots are counted, not only the removed spots.

#### Parameters

- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test.

## 3.2.6 MaxBadFeaturesRemoved

---

### Global maximum bad features removed test

#### Description

The aim of this global test is to count the number spotd which are rejected by all the unit tests. In this test only all removed spots are counted.

#### Parameters

- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test.

## 3.2.7 HeterogeneousFeature

---

### Heterogeneity feature test

#### Description

The aim of this test is to reject all spots which are too Heterogeneous. This Doelan test use the standard deviation of the green foreground (Genepix **F635 SD** column) and the standard deviation of the red foreground (Genepix **F532 SD** column).

#### Parameters

- **Maximal value of standard deviation:** The maximal value of the standard deviation for a spot to be not rejected.
- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

#### Graphics

This test will generate two graphics (one for each color) showing the distribution of the standard deviation of the intensities of the spots.

## 3.2.8 SaturatedPixels

---

### Saturated pixels feature test

#### Description

The aim of this test is to reject all spots with too many saturated pixels. This Doelan test use the percent of 635 foreground intensity (Genepix **F635 % Sat.** column) and the percent of 635 foreground intensity (Genepix **F532 % Sat.** column).

#### Parameters

- **Threshold maximum of saturated pixels:** The maximal value in the percent of saturated pixels for a spot to be not excluded.
- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

## 3.2.9 MinimalIntensity

---

### Minimal Intensity feature test

#### Description

The aim of this test is to reject all spots with a too low intensity. This Doelan test use the 635 foreground intensity (Genepix **F635 Median** column) and the 532 foreground intensity (Genepix **F532 Median** column).

#### Parameters

- **Minimal spot intensity:** The minimum value of the intensity in each color for a spot to be not excluded.
- **Channel to test:** The channel(s) to test (red, green or both).
- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

#### Graphics

This test will generate two graphics (one for each color) showing the distribution of intensities of the spots.

## 3.2.10 MaxDiameterFeature

---

### Maximal diameter feature test

#### Description

The aim of this test is to reject all spots with a too large diameter. This Doelan test use the Genepix **Dia.** column.

#### Parameters

- **Maximal diameter value:** The maximal value of the diameter in micrometer for a spot to be not excluded.
- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

#### Graphics

This test will generate a graphics showing the distribution of diameter of the spots.

## 3.2.11 MinDiameterFeature

---

### Minimal diameter feature test

The aim of this test is to reject all spots with a too small diameter. This Doelan test use the Genepix **Dia.** column of the GPR.

#### Parameters

- **Minimal diameter value:** The minimal value of the diameter in micrometer for a spot to be not excluded.
- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

#### Graphics

This test will generate a graphics showing the distribution of diameter of the spots.



## 3.2.12 Generic

---

### Generic feature test

#### Description

This test is a versatile test. You can use it to test one of the columns which compose an GPR file.

#### Parameters

- **Column name:** The name of the column to be tested. The name must be the same as in a GPR file.
- **Test:** The type of the test (lesser than, equals, greater than...).
- **Test value:** A value to be tested against all the values of the selected column.
- **Maximal threshold of bad spots:** The threshold in percent of invalid spots to reject the test. In the count, empty and abscent spots aren't considered.
- **Remove bad spot from the output list:** If enabled, spots which don't pass the test will be removed in output arraylist file.

## 3.3 Writing your own test

---

### Writing your own test

This section how to writing a simple plugin. In this example, we will create a test which filter spots with flag value at notfound. There is only one requirement to produce a functional plugin: it must contains one or more (a plugin can have more than one test) class witch extends `QualityUnitTest()` and implements the `Module()` interface.

The `Module()` interface force the developer to describe the module he want to add to doelan.

The abstract `QualityUnitTest()` impose you to set the parameters (name, type and default value) of test and writing a `test()` witch do the job.

The end of this section show you the annotated code of a test filtering spots with flag value at notfound. You'll find the source of this plugin and scripts to build it in the `doc/plugin` directory of doelan.

```
// Import Unit test class
import fr.ens.transcriptome.doelan.algorithms.QualityUnitTest;

// Import Result test classes
import fr.ens.transcriptome.doelan.data.QualityUnitTestResult;
import fr.ens.transcriptome.doelan.data.QualityUnitTestResult.SummaryResult;

// Import Module management classes
import fr.ens.transcriptome.nividic.platform.module.AboutModule;
import fr.ens.transcriptome.nividic.platform.module.Module;
import fr.ens.transcriptome.nividic.platform.module.ModuleDescription;
import fr.ens.transcriptome.nividic.platform.PlatformException;

// Import of the classes for handling bioassays
import fr.ens.transcriptome.nividic.om.BioAssay;
import fr.ens.transcriptome.nividic.om.SpotIterator;

// Import of the classes for handling parameters in a module
import fr.ens.transcriptome.nividic.util.parameter.FixedParameters;
import fr.ens.transcriptome.nividic.util.parameter.Parameter;
import fr.ens.transcriptome.nividic.util.parameter.ParameterBuilder;
import fr.ens.transcriptome.nividic.util.parameter.ParameterException;
import fr.ens.transcriptome.nividic.util.parameter.Parameters;

/**
 * Plugin Sample Test.
 * This class define a test based on not found spot flags.
 * @author Laurent Jourdren
 */
public class PluginSampleTest extends QualityUnitTest implements Module {

    /**
     * Get the description of the module.
     */
}
```

```

* The aboutModule() method allow to describe the test for the
* user in the doelan interface.
* @return The description of the module
*/
public AboutModule aboutModule() {

    ModuleDescription md = null;
    try {
        md = new ModuleDescription("PluginSampleTest",
            "Plugin sample Test : Test not found spot flag for BioAssay");
    } catch (RomdePlatformException e) {
        getLogger().error("Unable to create the module description");
    }
    return md;
}

/**
* Set the parameters of the element.
* The defineParameters() method allow to describe the type,
* the defaults values and the range of all parameters of the test.
* If the user enter enter in Doelan an out of range parameter, the
* message will tell him that parameter is wrong.
* @return The defaults parameters to set.
*/
protected Parameters defineParameters() {

    try {

        final Parameter threshold = new ParameterBuilder().withName("Threshold")
            .withType(Parameter.DATATYPE_DOUBLE).withDescription(
                "Threshold for the test").withGreaterThanValue(0)
            .withDefaultValue("0.10").getParameter();
        final Parameter filterFlags = new ParameterBuilder().withName(
            "Filter flags").withType(Parameter.DATATYPE_BOOLEAN)
            .withDescription(
                "Filter invalid features in output arraylist file")
            .withDefaultValue("false").getParameter();

        final FixedParameters params = new FixedParameters();
        params.addParameter(threshold);
        params.addParameter(filterFlags);

        return params;

    } catch (ParameterException e) {
        System.err.println(e);
    }

    return null;
}

/**
* Test the quality of the bioassay.
* @param bioassay BioAssay to test
* @param arrayList The array list
* @param parameters parameters of the test
* The test() method is the main function of the test. There, the test will be
executed.
* @return A QualityObjectResultTest Object
* @throws PlatformException if an error occurs while executing
* the test.
*/

```

```

public QualityUnitTestResult test(final BioAssay bioassay,
    final BioAssay arrayList, final Parameters parameters)
    throws PlatformException {

    QualityUnitTestResult result = null;

    try {

        final boolean[] results = new boolean[bioassay.size()];
        final int[] flags = bioassay.getFlags();

        // Get the user parameters values
        final double threshold = parameters.getParameter("Threshold")
            .getDoubleValue();
        final boolean filterFlags = parameters.getParameter("Filter flags")
            .getBooleanValue();

        int countNotFound = 0;
        int countRealSpot = 0;

        // Iterate over the bioassay to test all the flags of the spots
        SpotIterator si = bioassay.iterator();

        while (si.hasNext()) {
            si.next();

            if (si.isEmpty() || si.isFlagAbsent())
                continue;

            // If the flag of spot is not found, the spot is set to false in
            // the results array and will not be in the output gal file
            if (si.getFlag() == BioAssay.FLAG_NOT_FOUND)
                countNotFound++;
            else
                results[si.getIndex()] = true;

            countRealSpot++;
        }

        final double ratio = ((double) countNotFound) / ((double) countRealSpot);

        // Set the result information in the result variable
        result = new QualityUnitTestResult(bioassay, this);
        result.setMessage("Not found flag features : " + countNotFound + " / "
            + countRealSpot + " max : " + (countRealSpot * threshold));

        result.setGlobalResultType(true);
        if (filterFlags) result.setNewFlags(results);

        SummaryResult rac = result.getResultAllChannels();
        rac.setPercent(true);
        rac.setThresholdEqualityType("<=");
        rac.setUnit("%");
        rac.setThreshold(threshold);
        rac.setValue(ratio);
        rac.setPass(ratio <= threshold);

    } catch (ParameterException e) {
        throw new PlatformException("Error while creating parameters ("
            + this.getClass().getName() + ") : " + e.getMessage());
    }
}

```

```

        // Return the result
        return result;
    }

    //
    // Constructor
    //

    /**
     * Public constructor.
     * @throws PlatformException If the name or the version of the element
     *         is null.
     */
    public PluginSampleTest() throws PlatformException {
        // MUST BE EMPTY
    }
}

```

## Compiation and installation of plugins

This section presents you the process of building a fonctionnal plugin after writing the source code. All the next steps are very common, you can easily script it (with ant, unix or Windows shell). In the next chapter you'll find two scripts (one for \*nix and one for Windows) to automate the building of a simple plugin. Writing an Ant script will be useful if you consider to develop a large plugin.

- In your source directory compile your plugin. You can use a makefile like Ant to automate the compilation of your sources.
- Pack the generated *.class* file in a *.jar* file.
- Copy your *.jar* in the Doelan *plugins* directory.
- Restart Doelan.
- Now, You can use your own test in the application.

## Building the plugin sample

This an script for building the *.jar* file of the plugin under \*nix. Don't forget to replace *PATH\_TO\_DOELAN* by the valid path to Doelan on your filesystem in the script.

```

#!/bin/sh

DIRLIB=$PATH_TO_DOELAN/lib

for LIB in `ls $DIRLIB`
do
    CLASSPATH=$DIRLIB/$LIB:$CLASSPATH
done

rm *.class *.jar
javac -classpath $CLASSPATH *.java
jar cf testplugin.jar *.class

```



This is the same script for Windows. It's a batch file, save into a text file and rename it with the *.bat* extension. Be careful under Windows, with the *CLASSPATH* and *PATH* environment variable, sometimes they are not set. See your Java documentation for more information.

```
del *.class
del *.jar
javac -classpath %CLASSPATH% *.java
jar cf testplugin.jar *.class
```

## 3.4 PDF documentation

---

## 4.1 Get the source

---

### Download Doelan source

Doelan source is distributed in several formats for your convenience.

- [.tar.gz archive](#) (1.5 MiB)
- [.zip archive](#) (1.7 MiB)

Doelan is distributed under the [General Public License](#) .

### Compilation

You'll find more information about how compile Doelan in the [compiling section](#) .



## 4.2 Compiling

---

### How to compile Doelan

To build Doelan, you need a [Java 2 Standard Edition SDK](#) greater or equals to 1.4.x and [Maven](#) 1.x.

First set your current directory to the base directory of **Doelan**. The next commands works whatever your operating system and the "\$" character represent the command line prompt. Then, you can show all available commands of **Maven** with:

```
$ maven -g
```

To build build binary distribution:

```
$ maven dist-bin
```

To build build source distribution:

```
$ maven dist-src
```

To build build both source and binary distribution:

```
$ maven dist
```

To clean the build directories:

```
$ maven clean
```

## 4.3 Known bugs

---

### Known Bugs for Doelan 1.0

- Remove freezing actions from applet mode. It is a Genepix Bug. Actually, Axon doesn't support embedded java applets in Genepix.
- Graphics in Doelan are not antialiased. Due to a bug in Sun Java Virtual Machine, antialiasing isn't turn on. If you enable it, a fatal error sometimes occurs.

### Submit a Doelan bug

You can submit a bug by sending a sending a mail to the [doelan list](#) .

## 4.4 Futur improvements

---

### Futur improvements

- Load TestSuites from different source (not only filesystem, i.e. website, database) using the plugin system.
- Create several plugins for the export of the report (not only an html file, i.e. xml file, database).
- Create a new test: Test if signal/background ratio is lower than 15%
- Create a new test: Test if signal/background ratio is greater than 2 SD

## 4.5 Contribute to Doelan

---

### How can I contribute to Doelan?

Doelan is a free software project. Everybody is encouraged to contribute to Doelan. There are lots of ways in which you can help to continue Doelan:

- Testing new versions
- Submitting bug reports to the mailing list
- Send your TestSuites (with comments) to the developer to publish it on the website
- Programming new Doelan plugins

There are several places to look for more information if you want to get involved in the development. The first step is to subscribe to the mailing list. You will soon see something that can be improved or added.

### How do I commit a bug report?

There is actually no automatic bug tracking system available like Bugzilla for Doelan. So, to report a bug, you must send your report on the [mailing-list](#).

### Publish TestSuites and plugins on Doelan website

We can post on the [mailing-lists](#) your TestSuite suggestions for publication on website. For the pluggins, please contact the [developer](#) without sending your plugin in attachment.

## 5.1 Additional TestSuites

---

### Additional TestSuites

In this section you will find soon, samples TestSuites for Doelan with description. You can also publish your own TestSuite on Doelan website by send your creation on the [mail lists](#) .

## 5.2 Additional Test Plug-ins

---

### Additional Plugins

In this section you will find soon, test plugins for Doelan. You also publish your own plugins on Doelan website by send your creation on the [mail lists](#) .

## 5.3 Mailing list

---

### Mailing Lists

These are the mailing lists that have been established for this project. For each list, there is a subscribe, unsubscribe, and an archive link.

List Name	Subscribe	Unsubscribe	Archive
doelan-list	<a href="#">Subscribe</a>	<a href="#">Unsubscribe</a>	<a href="#">Archive</a>