CELLMISSY MANUAL

INTRODUCTION

This document is intended as a supporting material for researchers that wish to use the **CellMissy** tool.

CellMissy is a cross-platform, generic and easily extensible data management and analysis system for cell migration experiments, being focused in its current version on wound healing-like experiments. It is entirely written in Java and is freely available under the Apache2 open source license at https://cellmissy.googlecode.com/.

CellMissy is described in "*CellMissy: a tool for management, storage and analysis of cell migration data produced in wound healing-like assays.*" (P. Masuzzo, N. Hulstaert, L. Huyck, C. Ampe, M. Van Troys and L. Martens).

1. HOW TO RUN CELLMISSY FOR THE FIRST TIME

In its simple, single-user setup, **CellMissy** can run on minimal hardware, so any modern laptop or desktop PC is more than sufficient. Furthermore, since **CellMissy** is written in Java, it can run on any platform that supports a Java Virtual Machine version 1.7.0 or above (Windows, Linux, and Mac OS-X). However, if **CellMissy** is to be used as a shared system between many different users, it will be more practical to set up a central database (DB) that all users can access simultaneously. This DB server again needs not be a high-end machine, and the task can easily be handled by any modern desktop machine with sufficient storage space. **CellMissy** handles both scenarios (single-user or multi-user) with equal ease, and has been designed to support a full-blown production environment in a large lab without hiccups.

Before you can correctly use **CellMissy** for the first time, you need to follow some configuration steps, in order to set up a DB connection and create the MySQL schema for the application:

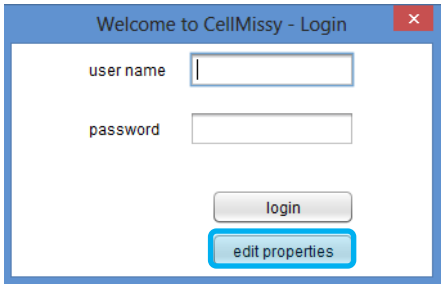
1. connect to a MySQL server (e.g. with MySQL Workbench, see http://www.mysql.com/products/workbench/)
2. create a new schema in the connected server and set it as the default schema (i. e. make the schema the active one in the current session)

3. run the SQL script "***cellmissy_schema.sql***" (you can download it from https://code.google.com/p/cellmissy/downloads/list): this will create the tables for the DB, set the indexes for them, and insert basic records into the DB (e.g. some cell lines, migration/invasion assays, extracellular matrix compositions...)

Please note that the provided SQL script works for MySQL relational databases; if you want to use different DB types, let us know and we'll try to provide you with another script.

After you have configured the DB connection for **CellMissy**, you can run the application by double clicking the executable *.jar* file present in the "***CellMissy***" folder (note that you need to unzip the compressed CellMissy folder before you can execute the *.jar* file). At this stage, you can edit **CellMissy** properties that establish the connection to the DB, by clicking the "*edit properties*" button in the login dialog, as shown below.

Welcome to CellMissy

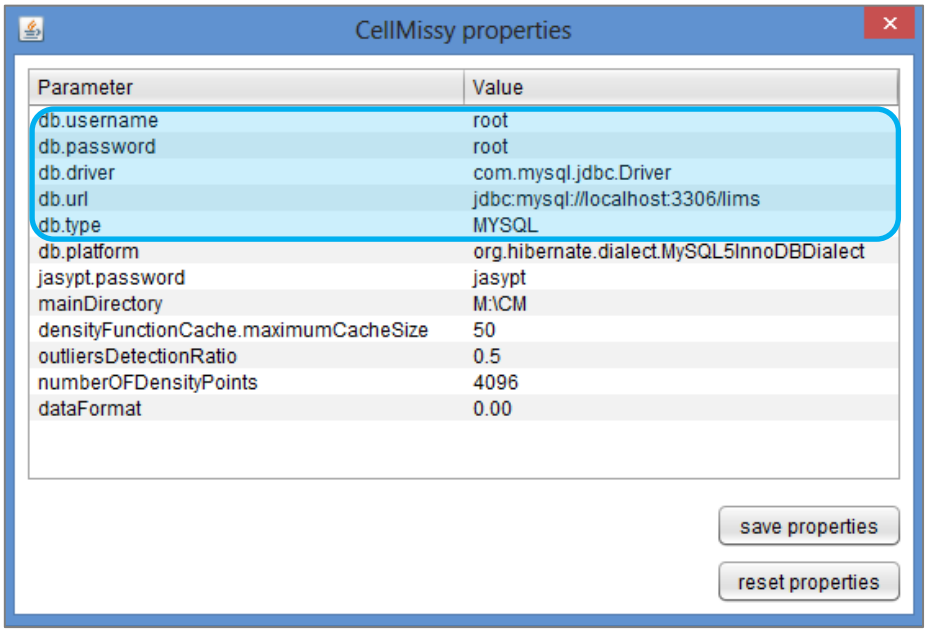


Another dialog will pop up with a table containing the properties of the software and their values, as shown in the screenshot below. Here, you can set the connection parameters (*db.username*, *db.password*, *db.driver*, *db.url*, *db.type*) according to the configuration properties you have chosen in the three steps above. Once the new properties have been saved, the application will automatically shut down: you can then restart **CellMissy** (again double clicking the executable *.jar* file) and use it with the current DB parameters.

To login into the application, you can use the credentials from the *root application user* (this *root application user* is inserted into the DB when you run the "***cellmissy_schema.sql***" script):

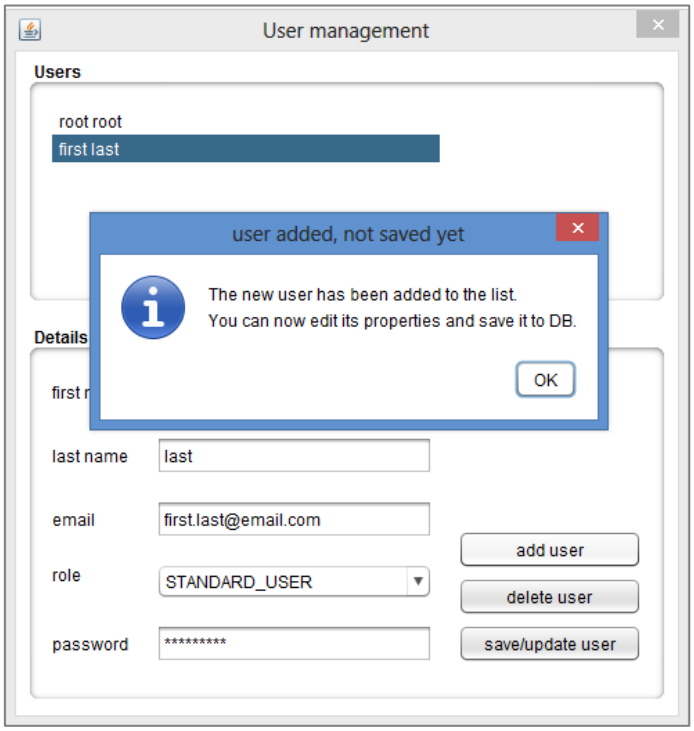
username	**password**
root	*cellmissy*

CONFIGURING CellMissy – CellMissy properties



Having logged in with this root user, you will have *ADMIN* rights: this means that you will also have access to the ***User Management*** module (through the *Edit* menu); here, you can delete the root user or change its credentials, and you can add other application users as well. Each time you add a user, this is automatically inserted in the list present in the GUI (see following screenshot), you can then edit the data of the new added user and finally save the user to the DB. Note that if you enter the application as an *ADMIN* user, you can also select a user from the list and delete it from the DB.

USER MANAGEMENT in CellMissy



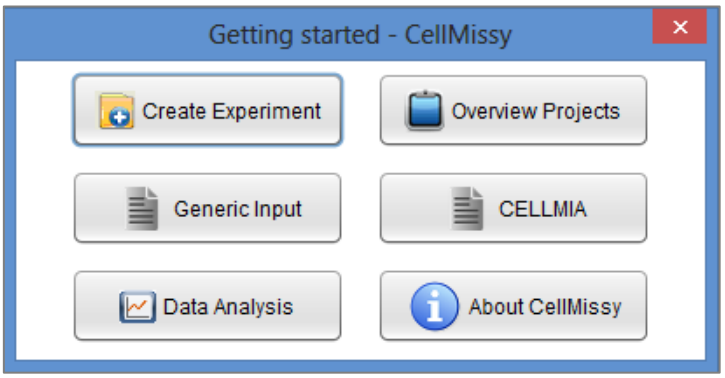
2. HOW TO USE CELLMISSY

Developed to follow the steps typically encountered in a cell migration experiment, **CellMissy** is mainly composed of three modules, as described below.

- **Experiment Manager** → to set up a new cell migration/invasion experiment.
- **Data Loader** → to import and store cell migration data; for a typical wound healing-like experiment these are values of measured area in time.
- **Data Analyzer** → to explore and analyze cell migration data; perform statistics, and finally create analysis reports.

Each of these main modules can be accessed in **CellMissy** through the main *File* menu, as well as through the start-up dialog that will automatically appear when you run **CellMissy**, as shown in the next screenshot.

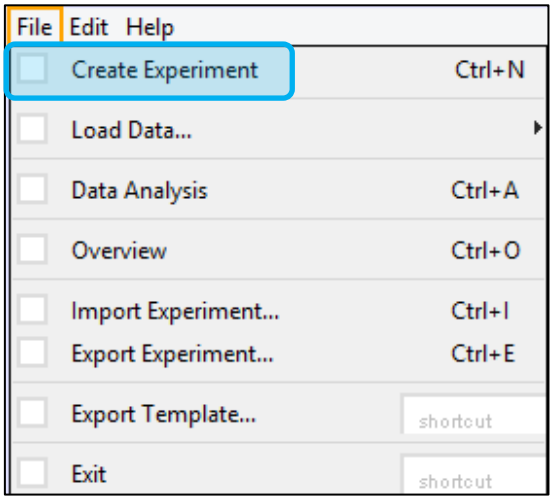
Getting started with CellMissy



On top of these modules, **CellMissy** also provides means for import/export of an entire experiment, along with the import/export of templates containing all the experimental set-up metadata. These functionalities can be accessed through the *File* menu.

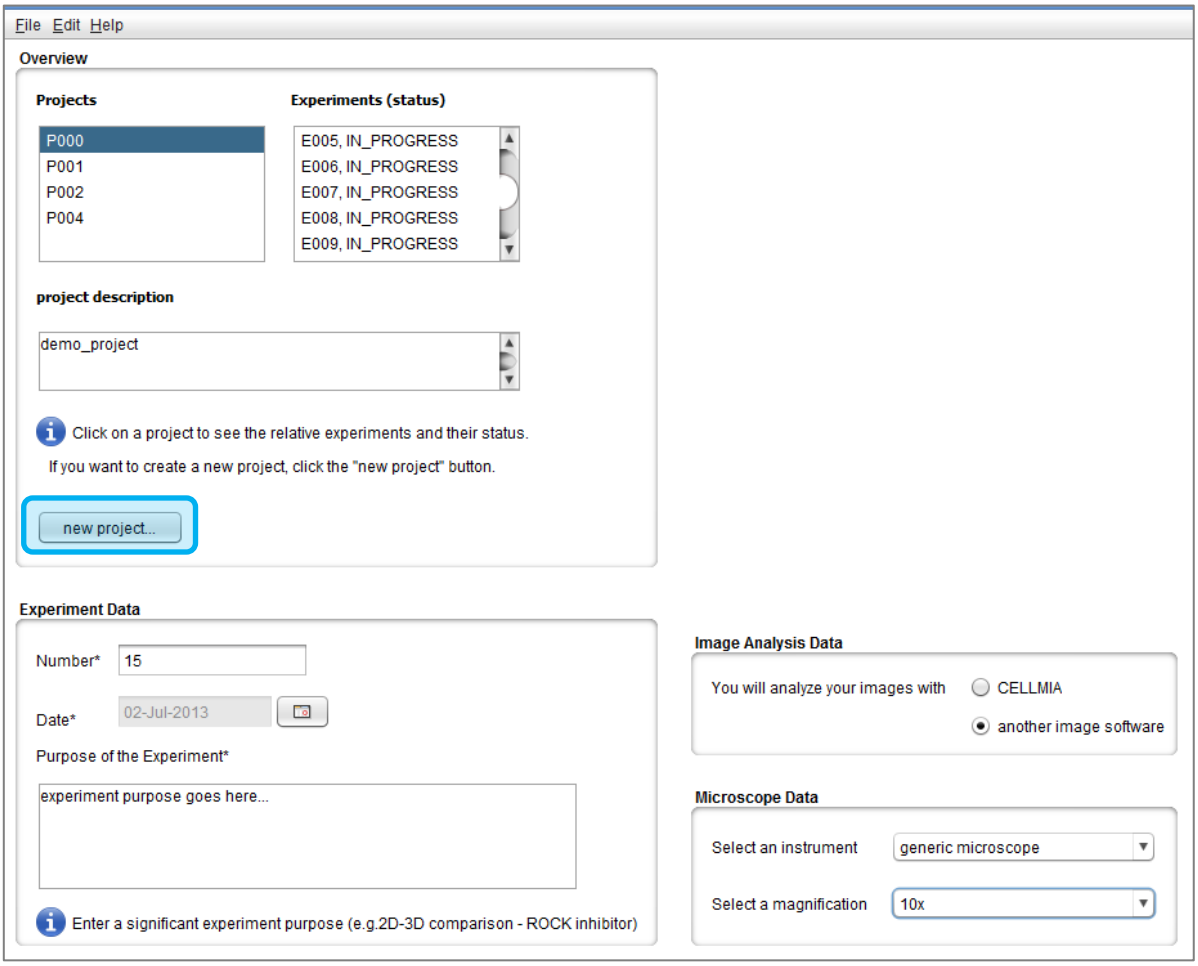
The next sections of this document will provide more details on each of the module of **CellMissy**.

2.1 EXPERIMENT MANAGER - CREATE A NEW CELL MIGRATION EXPERIMENT



This module guides you through the set-up of a new cell migration/invasion experiment.

EXPERIMENT MANAGER – experiment metadata

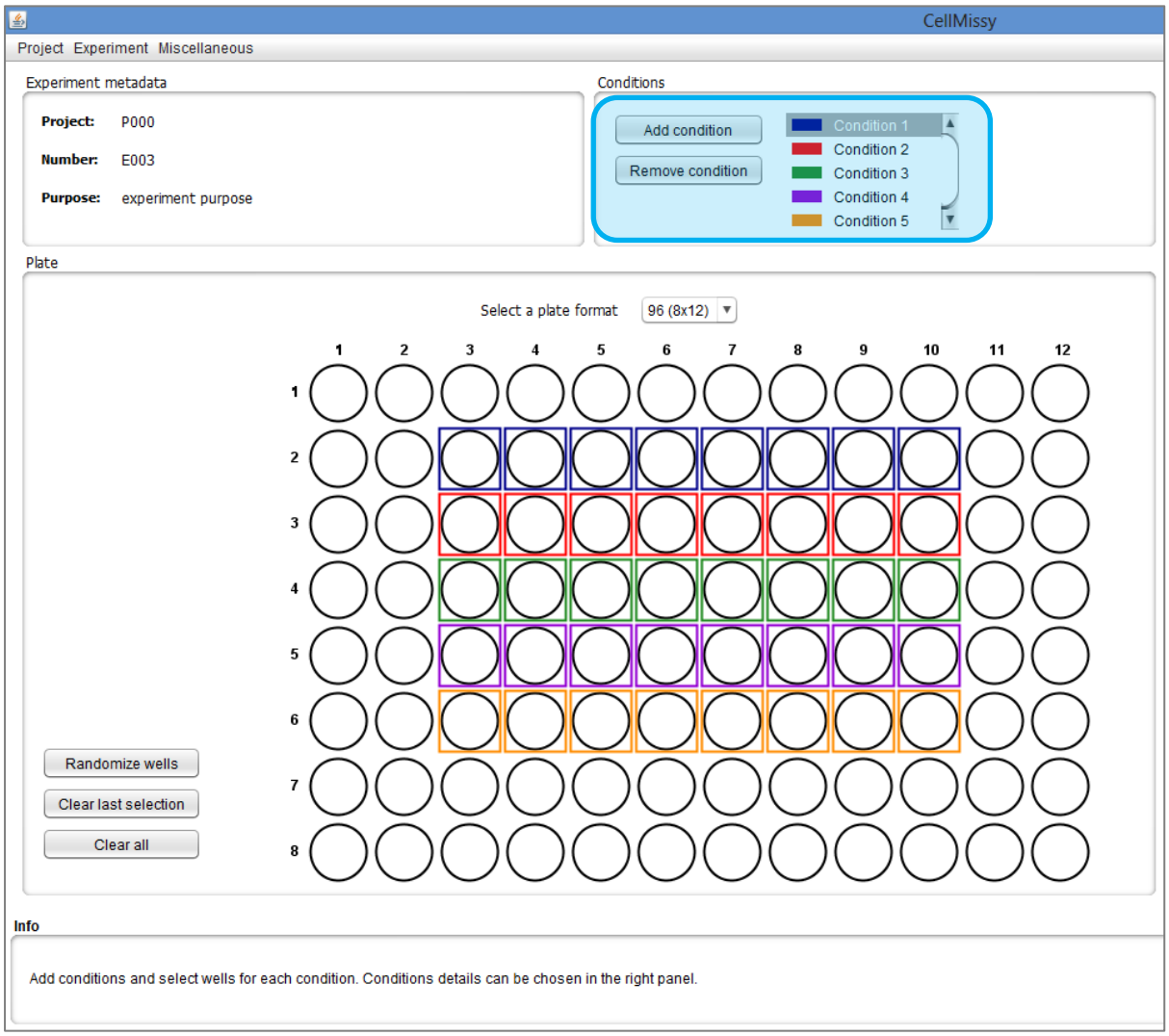


You first choose a project to which the experiment is going to belong, and provide a number and a short description for the experiment (see the screenshot above, experiment metadata). If the

experiment you want to add needs to be part of a new project (i.e. a project that is not present in the DB yet), you can add a new project from this interface.

Then, you can define the experimental set-up on a multi-well plate view (see the screenshot below, plate view and biological conditions). Common multi-well plate formats are rendered on the view, namely 96 (8 columns x 12 rows), 48 (6 x 8), 24 (4 x 6), 12 (3 x 4), 6 (2 x 3).

EXPERIMENT MANAGER – plate view and biological conditions



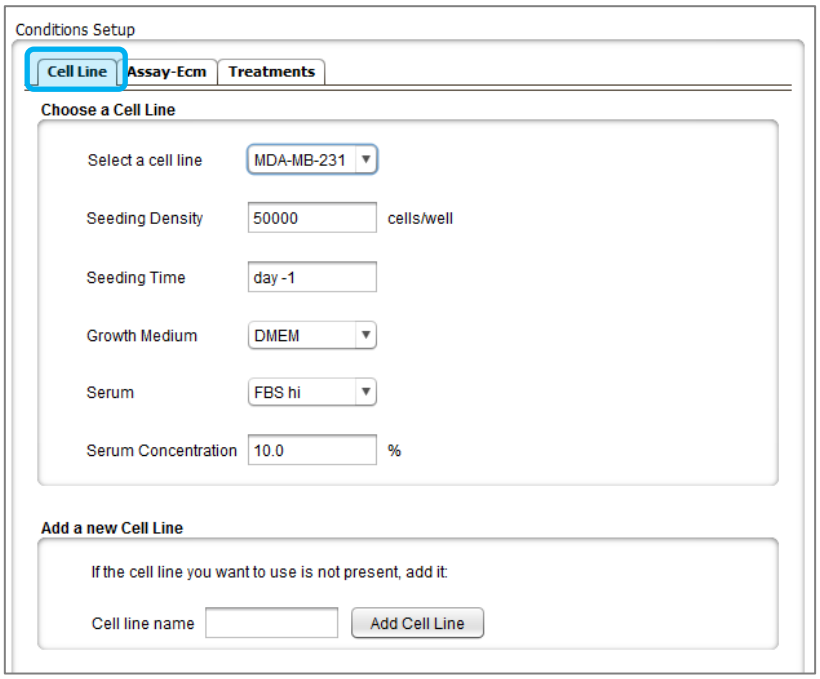
Different informative metadata variables can be added or chosen from drop down lists for each biological condition in the right panel of the GUI. Here, three tabs show different views according to the variables to be supplied (see the next three screenshots):

1. **Cell Line**: here the cell line used in the biological condition can be chosen; you can define and customize parameters such as seeding density (expressed in number of cells per

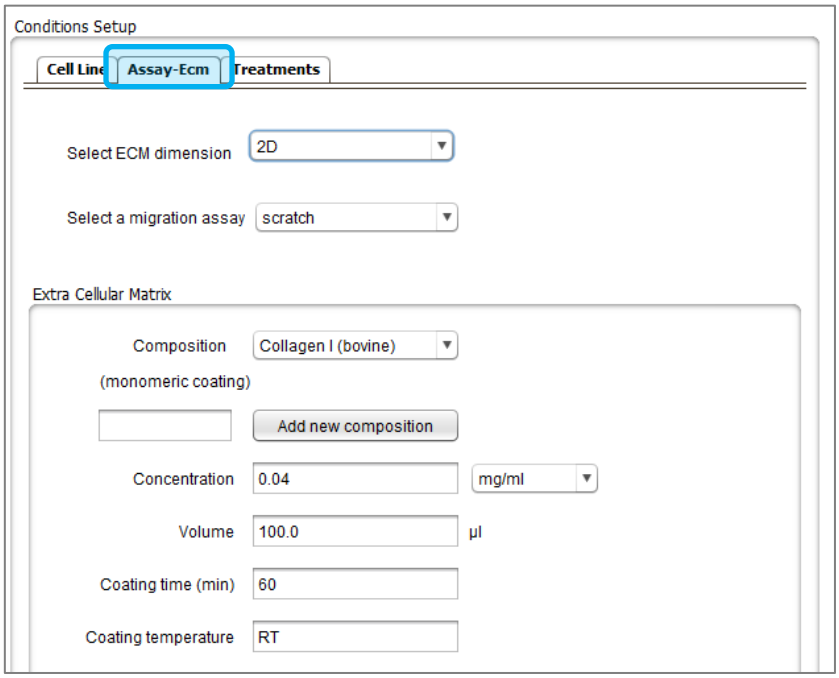
well), growth medium and serum type and concentration. Cell lines can be added to a drop down list (and thus to the used **CellMissy** DB).

2. **Assay_ECM**: here you can characterize the extracellular matrix (ECM) condition, the dimensionality (are you setting up a migration (2D) or an invasion (3D) experiment?), the coating type (collagen, fibronectin), coating/matrix polymerization conditions (temperature, time…) and so on.
3. **Treatments**: the last tab gives you the possibility to specify the treatment or compound to which the cells were subjected in a given biological condition, for example the type and concentration of a drug, the presence of only drug solvent (e.g. in control condition), a protein overexpression or a siRNA treatment, etc. Treatments can be added to a drop down list (and thus to the used **CellMissy** DB).

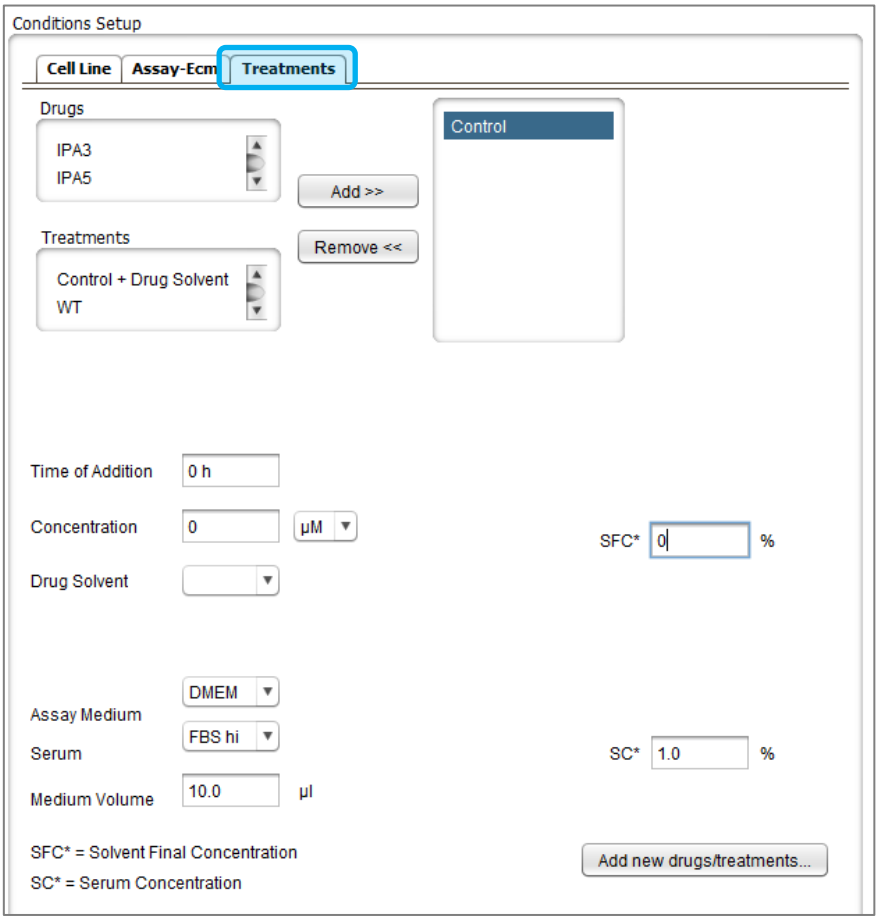
EXPERIMENT MANAGER – cell line properties



EXPERIMENT MANAGER – assay-ECM properties



EXPERIMENT MANAGER – treatments properties

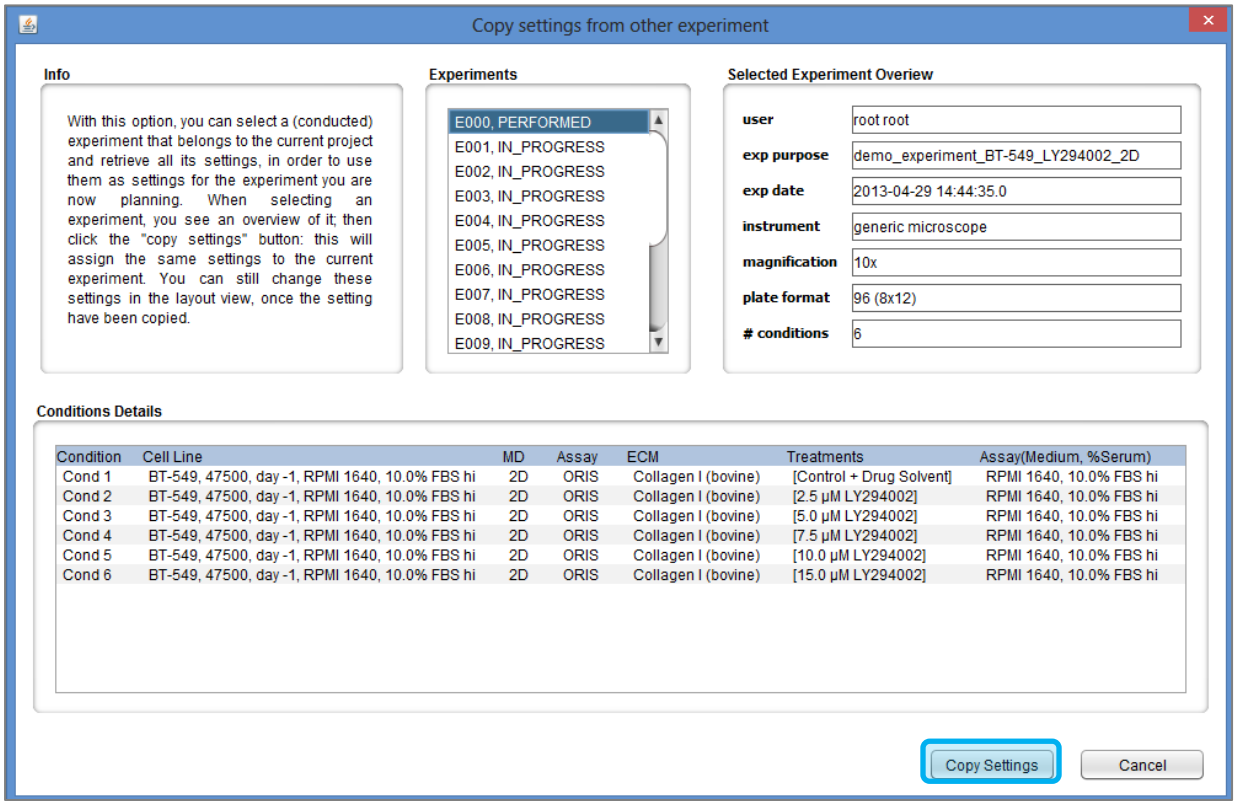


Once this experimental set-up is finalized, you can export the design as a PDF document that can be used as a reference in the lab while conducting the experiment. Then, the plate lay-out, well assignments and related metadata are all stored in the database.

2.1.1 EXPERIMENT MANAGER – IMPORT SET-UP SETTINGS FROM ANOTHER EXPERIMENT

CellMissy provides the possibility, while setting-up a new experiment, to retrieve the settings from an experiment that was already planned (thus present in the DB) for the current project. You can easily use this functionality in the Experiment Manager module, through the "*Import Settings…*" button. Clicking this button will make a dialog appear, where you can select the experiment from which you want to retrieve the settings, as shown in the following screenshot. Clicking an experiment will render its details in the right panel, while a table with all the biological conditions details will be shown at the bottom of the dialog. Clicking the "C*opy Settings*" button will assign the selected experiment's set-up to the currently planned experiment, and this will automatically update the plate lay-out, as well as the conditions list.

EXPERIMENT MANAGER – importing settings



2.1.2 EXPERIMENT MANAGER – EXPORT A TEMPLATE TO AN XML FILE

Once the experimental set-up is finalized, just after you have created a PDF report, you can export the set-up of an experiment to an XML file, creating thus a template that can be exchanged and easily re-imported into **CellMissy** to reproduce someone else's workflow/set-up. This can be done in the Experiment Manager module clicking the "*Export Template…*" button: all you have to do is choose a directory to save the XML file and click the "*save*" button. A name for the template file will be automatically created by **CellMissy**, of the type: *setup_template_EXXX_PYYY*, with *X* and *Y* to be replaced with the number of the experiment and the number of its project, respectively.

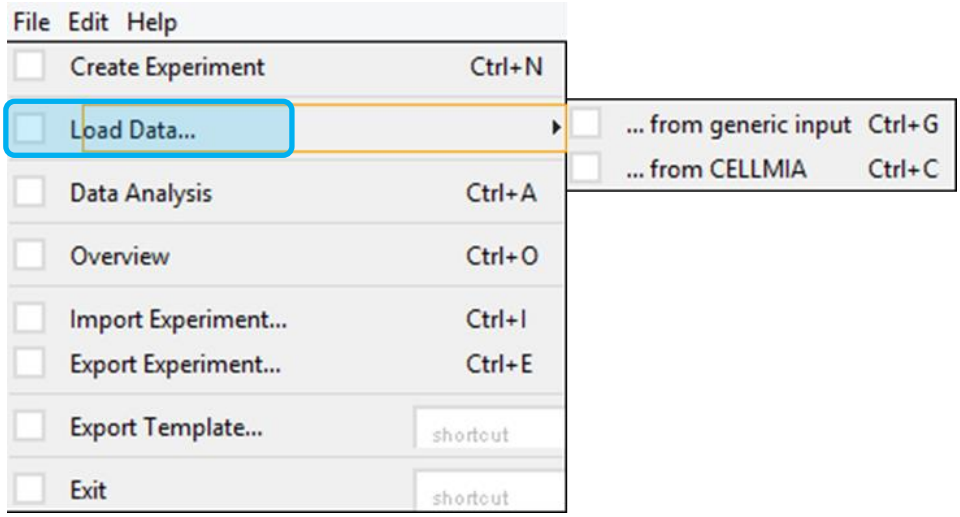
At https://cellmissy.googlecode/downloads/list you can find an example of this set-up template (the *setup_template_E000_P000.xml* file), along with the XML schema of **CellMissy** (the *cellmissySchema.xsd* file).

The same function can be reached in **CellMissy** also through the *File* menu: in this case, you first need to choose an experiment from which you want to export the template, and then save the generated file to a directory.

2.1.3 EXPERIMENT MANAGER – IMPORT A TEMPLATE FROM AN XML FILE

What if you want to use the settings of an experiment that was not saved in your **CellMissy** database? If this is the case, during the set-up of a new experiment, you can import an external XML file containing the experiment template (a file generated in **CellMissy** as described in section 2.1.2 above). In the Experiment Manager, clicking the "*Import Template…*" button will let you choose an XML file to import; **CellMissy** will then retrieve the settings from this file and assign them to the new experiment. The plate lay-out and the conditions list will be then automatically updated.

2.2 DATA LOADER - LOAD MIGRATION DATA…



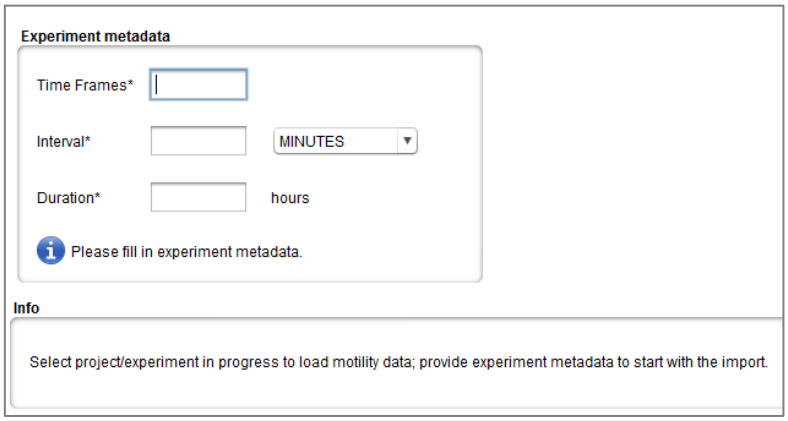
Once the experiment is performed and the acquired images are analyzed by the image processing software of your choice, you return to **CellMissy** to import the relevant cell migration data. Data import and storage in **CellMissy** are either based on using the generic migration input format (requiring you to connect text files with a replicate (i.e. a well in the experimental set-up) of a specific biological condition) or can be fully automated once tailored to a customized system. See the two following sections for more details.

2.2.1 … from generic input

A wound healing-like experiment followed by image processing will generally result in a list of area values in time for each technical replicate of a biological condition. This can be for a limited number of time points or for a large number (e.g. based on a time-lapse experiment). The image processing will either have given you as output the area values of the wound/gap/cell-free zone (that decreases in time as cells in the sheet migrate) or the area of the cell-covered zone (that increases in time as cells in the sheet migrate). **CellMissy** is designed such that both types of area *vs.* time lists can be used as import data when present in a generic input file (see "*example_dataset_scratch*", see also below, Section 3).

For generic input, you first need to provide some experiment metadata: experiment duration, time interval, etc. (see next screenshot).

DATA LOADER – experiment metadata



Then, **CellMissy** expects for each well at least one tab-separated text file containing two columns, as shown in the following figure.

INPUT DATA FILE – example

	time	area
2	0	0
3	30	4.155276505
4	60	9.919203631
5	90	19.28003667
6	120	23.29390134
7	150	29.54343138
8	180	37.01400293
9	210	46.8975504
10	240	48.85572839
11	270	54.60291895
12	300	61.74627854
13	330	64.44038024
14	360	72.94498036
15	390	77.04713647
16	420	79.31239403
17	450	85.06734648
18	480	87.10290076
19	510	89.78463195
20	540	92.40887669
21	570	94.6932964
22	600	96.33930109
23	630	96.4307458
24	660	98.60261818
25	690	99.00041478
26	720	99.10034904

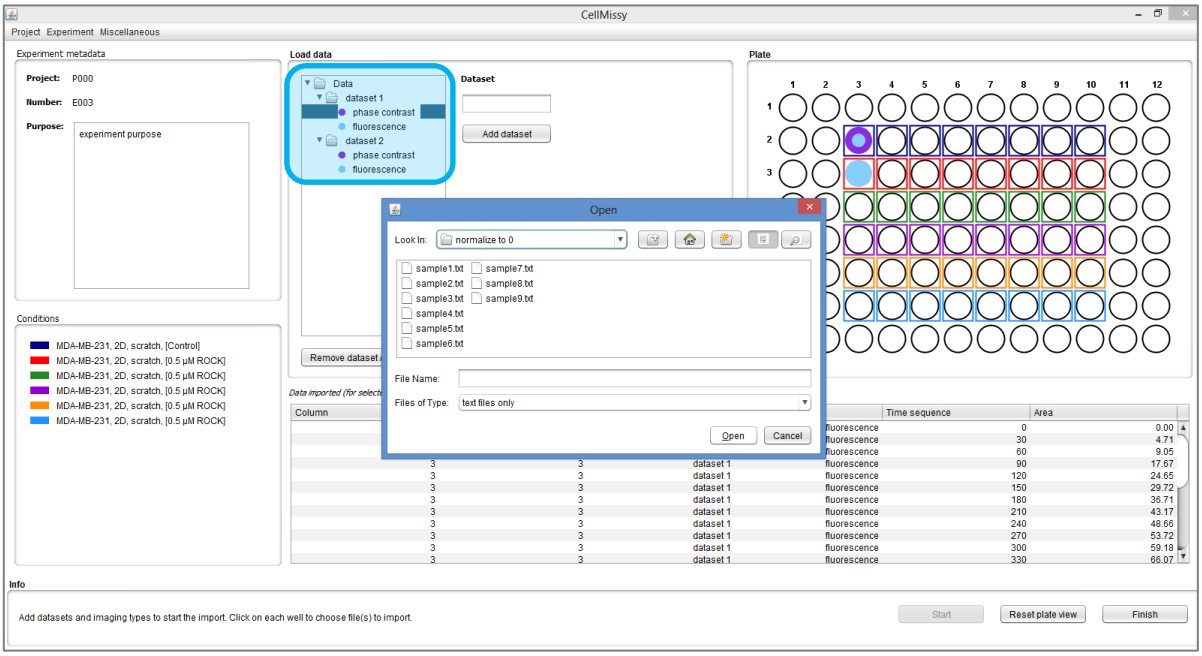
The first column holds the time information (this can be minutes or simply time sequence steps) and the second column the area values (these can be expressed in μm^2, pixels or area percentage %).

You start the import by adding names for datasets and imaging types of interest (e.g. phase contrast imaging, fluorescent imaging) (see screenshot below). One dataset could e.g. stand for a

collection of data generated through a specific setting of the algorithm during image processing. Once an imaging type is added to the session, it is automatically added to each dataset. However, a well (or a biological condition) may have been imaged with only a specific imaging technique, or a certain dataset may have been generated only for a specific group of conditions. That's why **CellMissy** does not require every combination of dataset/imaging type to be supplied for each well. Furthermore, you can load multiple files per well, even for a certain combination of dataset/imaging type, allowing the use of multiple imaging locations within the well.

A table underneath the plate view keeps track of the imported area values, together with current dataset, imaging type, and well's column, row. Once the import is done, and experiment metadata are supplied, you can store the data clicking the "*Finish*" button.

DATA LOADER – adding datasets/imaging types – selecting *.txt* files

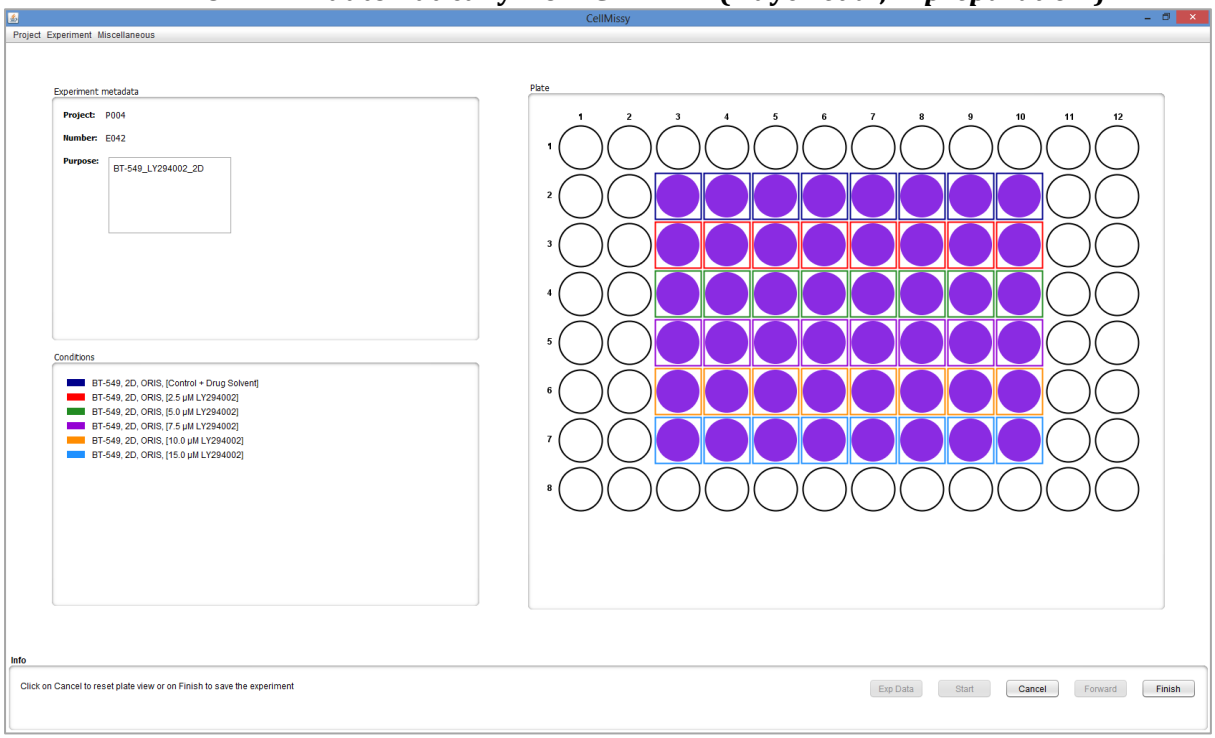


2.2.2 … automatically from a customized combination of imaging system and image processing software

CellMissy has capabilities to automatically load data (i.e. the text files generated by the image processing software) by reading metadata coupled to the images, to the processed images and, when provided, with information on the imaging order of the wells. This makes it amenable to high-throughput data processing. In the current version, this is established for the customized set-up used in our group, i.e. an *Olympus xcellence CellM* system and custom imaging software

we generated in collaboration with DciLabs Belgium (CELLMIA, Huyck et al. Shifting Quantitative Analysis of Migration Dynamics in 3D-Matrices to Higher Throughput, *in preparation*).

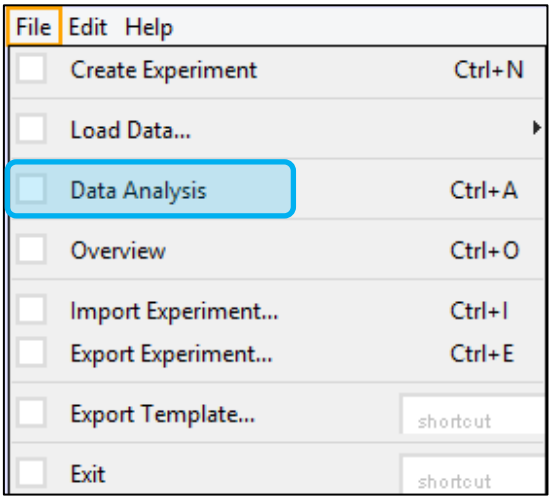
DATA LOADER – automatically from CELLMIA (Huyck *et al*, *in preparation.*)



Once you select the experiment for which you want to import and store the data, **CellMissy** looks for a file generated at microscope imaging time; this file has an *.obsep* extension (see http://loci.wisc.edu/software/bio-formats for further information), and contains the experiment metadata information, as well as the names and the metadata of imaging types used during the image recording and the names of the positions list(s) defined and generated. If the file is not in the right folder or more than one file is present, you can select a file to associate to the current session. Once these metadata are retrieved, **CellMissy** processes the data text files: all you have to do is to click on the well/sample that was first imaged; all the others wells will be automatically highlighted (if imaged), according to the position list associated to the current imaging type (see screenshot above). The plate view can be reset at any time and the import can be cancelled for a certain bench of data. Once the import is finished, you can save the cell migration data to the database clicking the "*Finish*" button.

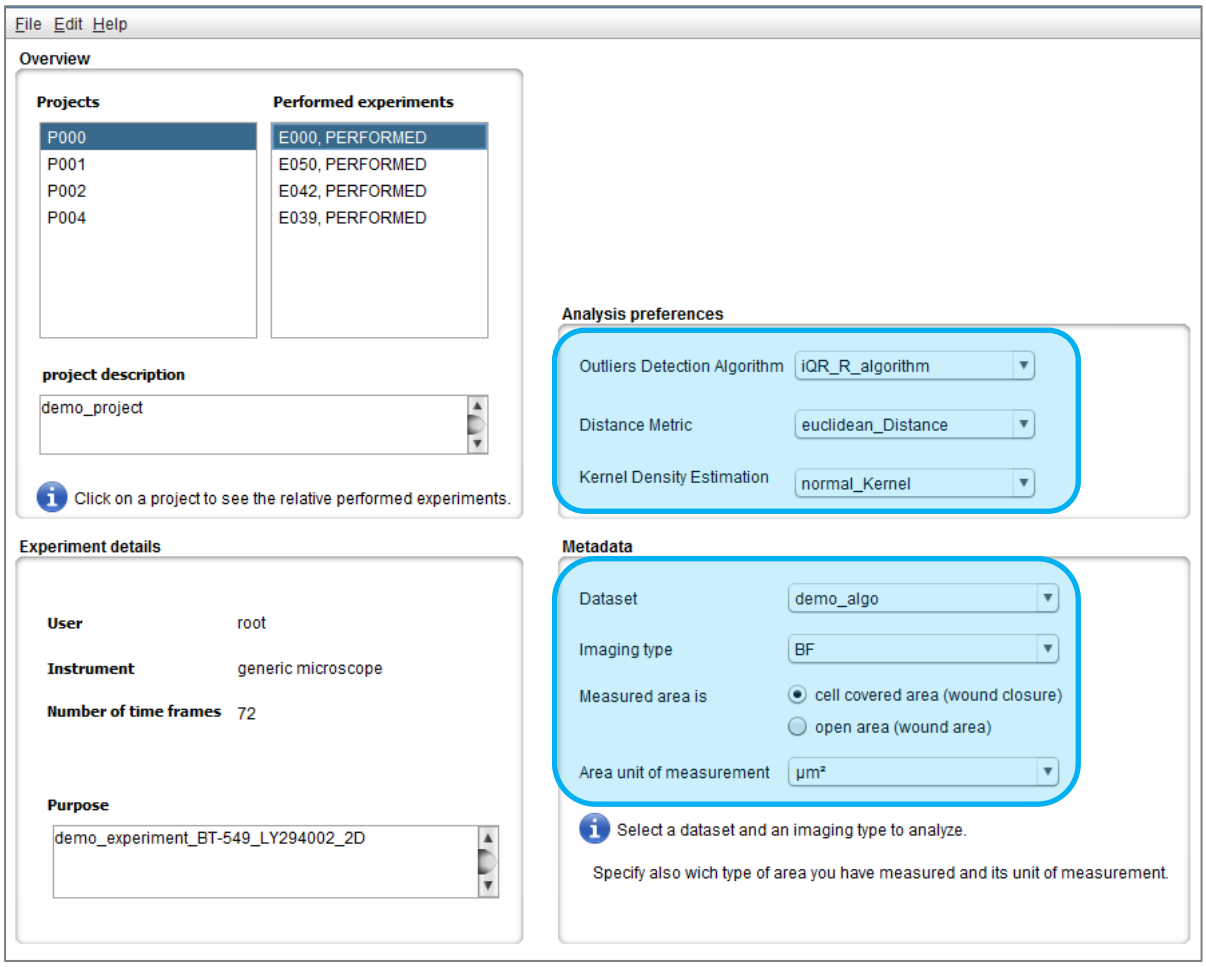
Of course, automated import from other existing image analysis tools will require the writing of a small piece of adapter code, and we foresee requests from the community to provide such interfaces, since we are fully committed to supporting such work, or undertaking it ourselves.

2.3 DATA ANALYZER – ANALYZE, EXPLORE AND REPORT DATA



Once cell migration data are loaded in the system, interpretation and reporting of these data can be performed by **CellMissy** in the Data Analyzer module.

DATA ANALYZER –analysis preferences/experiment metadata

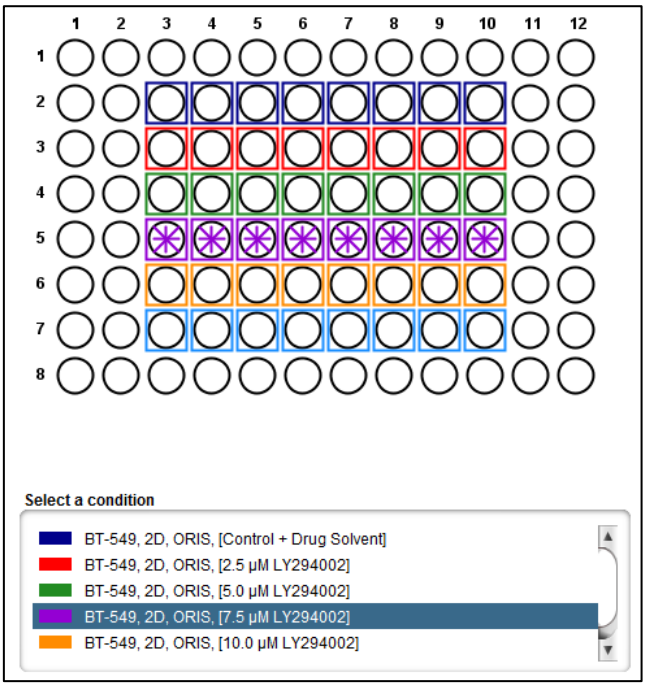


Here, you select an experiment in the main view (left side above screenshot), and a small summary of it is provided. For the chosen experiment, you need to select a dataset and an imaging type of interest: cell migration data associated with these two will be loaded from the DB, allowing data inspection and further analysis.

Before starting an analysis session, you need to provide the area unit of measurement (this can be μm^2, pixels or %), and specify whether the measured area in the wound healing-like assay is the cell-covered area (i.e. related to the wound closure) or the open area (i.e. the wound area). Furthermore, you can here select some analysis preferences: the algorithm to use for the outliers detection, the distance metric (for quality control on technical replicates level), and finally the kernel function for the probability density estimation. Note that every new available implementation for these analysis features will be automatically picked up by **CellMissy** and presented to you in its interface. For more details on the software extension, see section 2.5.

The general workflow for the data analysis is the same for both types of readouts to analyze, cell-covered area and open area, but open area values are first transformed to cell-covered area values and always expressed in area %.

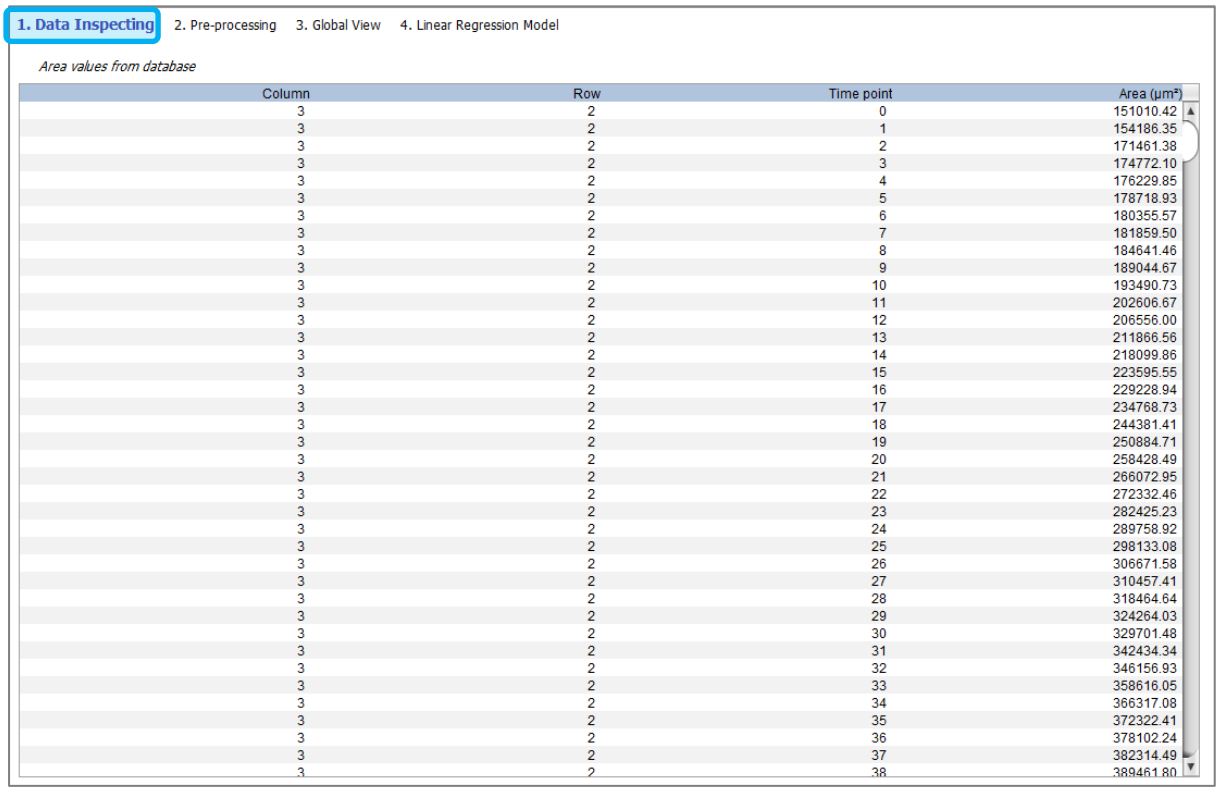
The first two steps in the data analysis are performed on the level of one biological condition: the left side of **CellMissy** interface is here composed of a plate lay-out showing the set-up of the experiment, along with a list reporting the annotated biological conditions (colors, followed by details). As shown in the following screenshot, when clicking on a biological condition, the correspondent replicates (wells) are marked with a star.



Then, area values are retrieved from the DB and thus explored through different steps, as detailed below and shown in the next screenshots.

1. **Data Inspecting**. The first view encountered is simply a data inspector, where the area values are shown, together with time information and well's coordinates ((column, row), e.g. (3, 2)).

DATA ANALYZER – inspecting the raw data (area values)

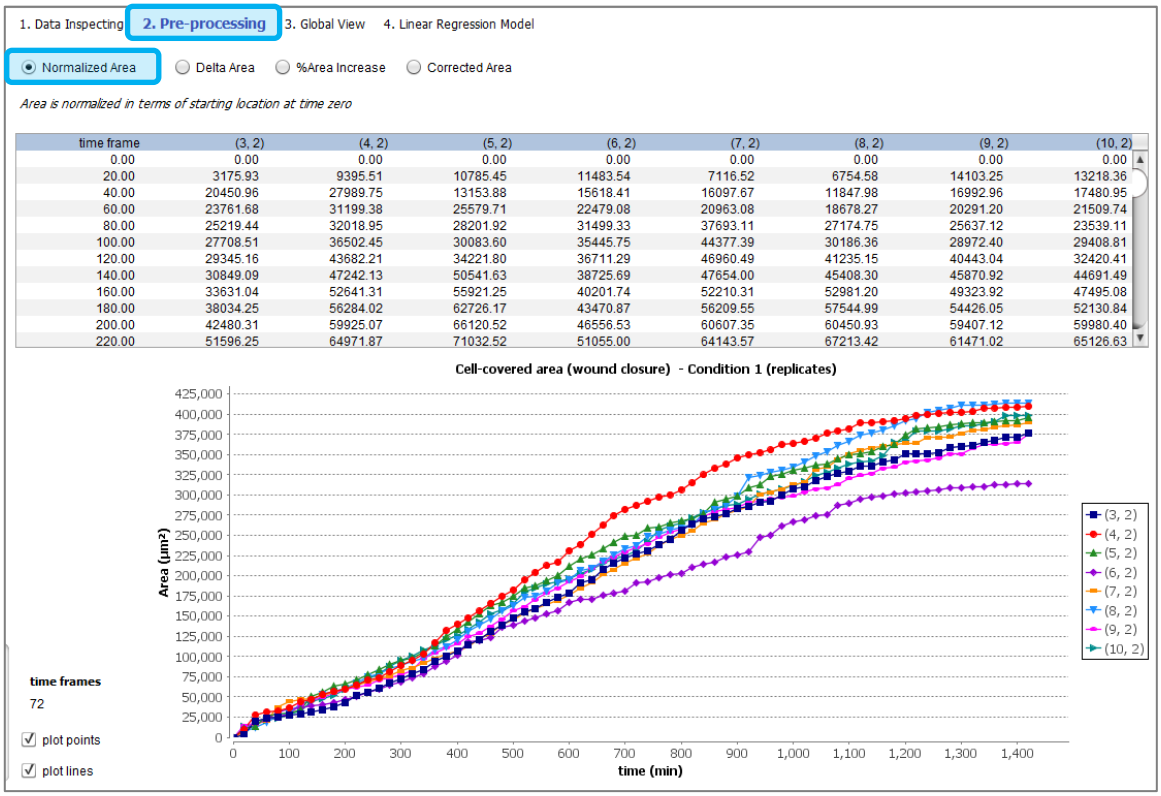


Column	Row	Time point	Area (µm²)
3	2	0	151010.42
3	2	1	154186.35
3	2	2	171461.38
3	2	3	174772.10
3	2	4	176229.85
3	2	5	178718.93
3	2	6	180355.57
3	2	7	181859.50
3	2	8	184641.46
3	2	9	189044.67
3	2	10	193490.73
3	2	11	202606.67
3	2	12	206556.00
3	2	13	211866.56
3	2	14	218099.86
3	2	15	223595.55
3	2	16	229228.94
3	2	17	234768.73
3	2	18	244381.41
3	2	19	250884.71
3	2	20	258428.49
3	2	21	266072.95
3	2	22	272332.46
3	2	23	282425.23
3	2	24	289758.92
3	2	25	298133.08
3	2	26	306671.58
3	2	27	310457.41
3	2	28	318464.64
3	2	29	324264.03
3	2	30	329701.48
3	2	31	342434.34
3	2	32	346156.93
3	2	33	358616.05
3	2	34	366317.08
3	2	35	372322.41
3	2	36	378102.24
3	2	37	382314.49
3	2	38	389461.80

2. **Pre-processing**. Under this second step pre-processing of the data is performed:

   - Normalization. The raw data are first normalized relative to the measured area at time zero. If you have loaded open area (wound/gap), the start area (time 0) is set to 100 and values are normalized to this, and then expressed in percentage of cell-covered area (complementary to 100). If you loaded cell-covered area, the start area is set to 0 and values are normalized to this. Here you also see the plot in time of the normalized area values for each technical replicate of the selected biological condition.

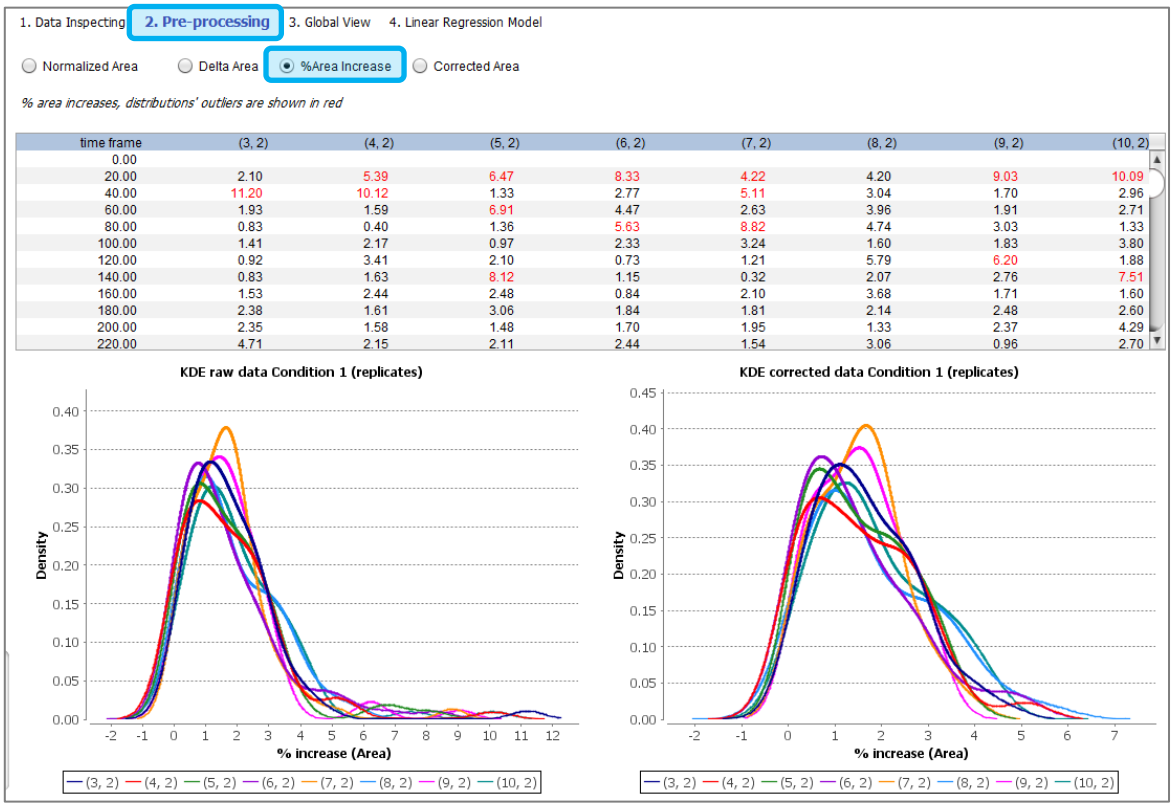
DATA ANALYZER – normalizing the area



time frame	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)	(8, 2)	(9, 2)	(10, 2)
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20.00	3175.93	9395.51	10785.45	11483.54	7116.52	6754.58	14103.25	13218.36
40.00	20450.96	27989.75	13153.88	15618.41	16097.67	11847.98	16992.96	17480.95
60.00	23761.68	31199.38	25579.71	22479.08	20963.08	18678.27	20291.20	21509.74
80.00	25219.44	32018.95	28201.92	31499.33	37693.11	27174.75	25637.12	23539.11
100.00	27708.51	36502.45	30083.60	35445.75	44377.39	30186.36	28972.40	29408.81
120.00	29345.16	43682.21	34221.80	36711.29	46960.49	41235.15	40443.04	32420.41
140.00	30849.09	47242.13	50541.63	38725.69	47654.00	45408.30	45870.92	44691.49
160.00	33631.04	52641.31	55921.25	40201.74	52210.31	52981.20	49323.92	47495.08
180.00	38034.25	56284.02	62726.17	43470.87	56209.55	57544.99	54426.05	52130.84
200.00	42480.31	59925.07	66120.52	46556.53	60607.35	60450.93	59407.12	59980.40
220.00	51596.25	64971.87	71032.52	51055.00	64143.57	67213.42	61471.02	65126.63

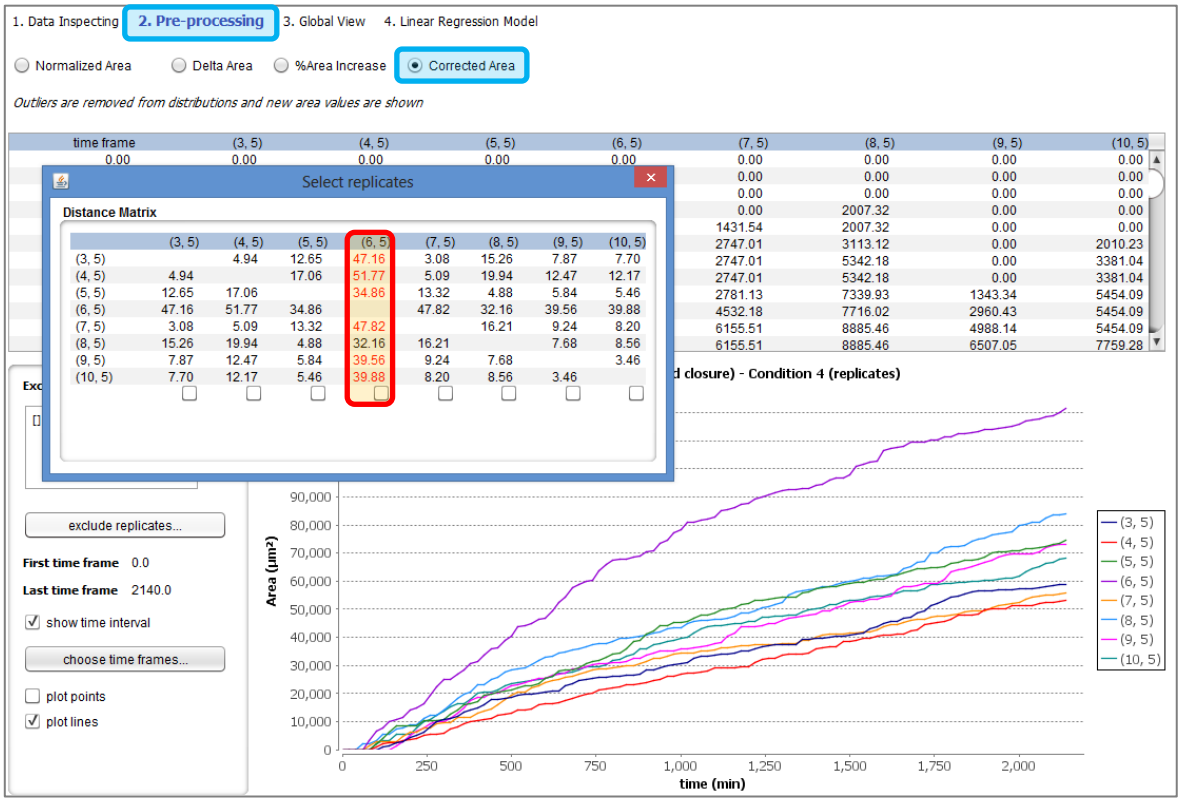
- <u>Data quality control</u>.
  - Delta area increments between time frame *t(n)* and *t(n+1)* are computed and shown, and percentages of area increase between consecutive time points are moreover calculated. Here, a first level of quality control takes place. For each well/replicate, area increases that are likely artifacts (possibly occurring due to experimental errors (e.g. cells or non-cell-objects in the cell-free zone) or due to false segmentation (e.g. when using fully automated image processing) are detected as outliers and corrected. This procedure is visualized using a Kernel Density Estimator (KDE) (see screenshot below, data quality control 1), plotting the probability density functions of all % area increases between consecutive time points for the different replicates. The left chart always presents raw data, and outliers are highlighted in red in the data table, while the right chart depicts corrected data. Note that this KDE is performed using a Gaussian kernel function, but **CellMissy** gives the possibility to extend this to other implementations.

DATA ANALYZER – data quality control 1: Kernel Density Estimation



- The corrected data are better visualized in this next step, where also the technical precision between replicates is examined (second level of quality control) using Euclidean distance (or any other distance metric you have selected in the analysis preferences list) as the similarity metric. The table here indicates for a specific replicate to what extent it resembles or deviates (indicated in red) from the other replicates in the biological condition.
- At both levels of quality control, you can either accept or decline the suggestion made by **CellMissy**. E.g. by unchecking the box beneath the replicate in the table you can decide to keep in a replicate designated as outlier (see screenshot below, data quality control 2).

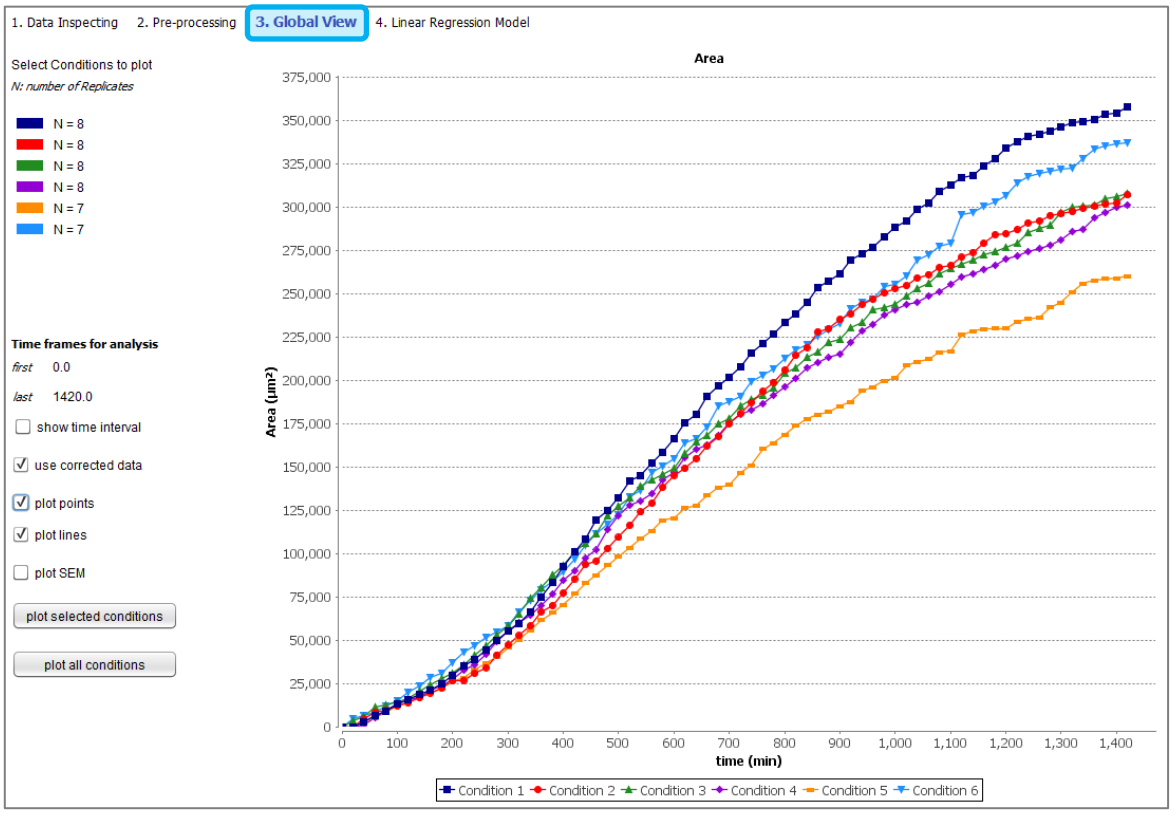
DATA ANALYZER – data quality control 2: Distance Matrix table



- Finally, you can adjust the time interval in which you wish to perform data analysis. Since data analysis implies the use of a linear regression model (see below), you optimally choose the time range where the area evolution is linear. You can change both the start and end time point (e.g. select the range 20-1000 minutes instead of 0-1600 minutes). Note that setting a limitation on the time range in one biological condition, automatically implies that only this time range will be considered in the subsequent analysis steps for all biological conditions in the experiment.

3. **Global view**. In the Global View step you pass from looking at one of the biological conditions to all the tested conditions (thus the entire experiment), therefore the list with the conditions is now disabled and you cannot interact with it anymore, unless you go back to previous steps of the analysis. You see now area over time plotted for each biological condition. This is based on plotting the median area across all the replicates for each biological condition. You can either use corrected data or retain the original raw data, and you can also select a subset of conditions you want to include into the plot. Standard error of the median can be plotted on top as well (by checking the corresponding box).

DATA ANALYZER – Global View

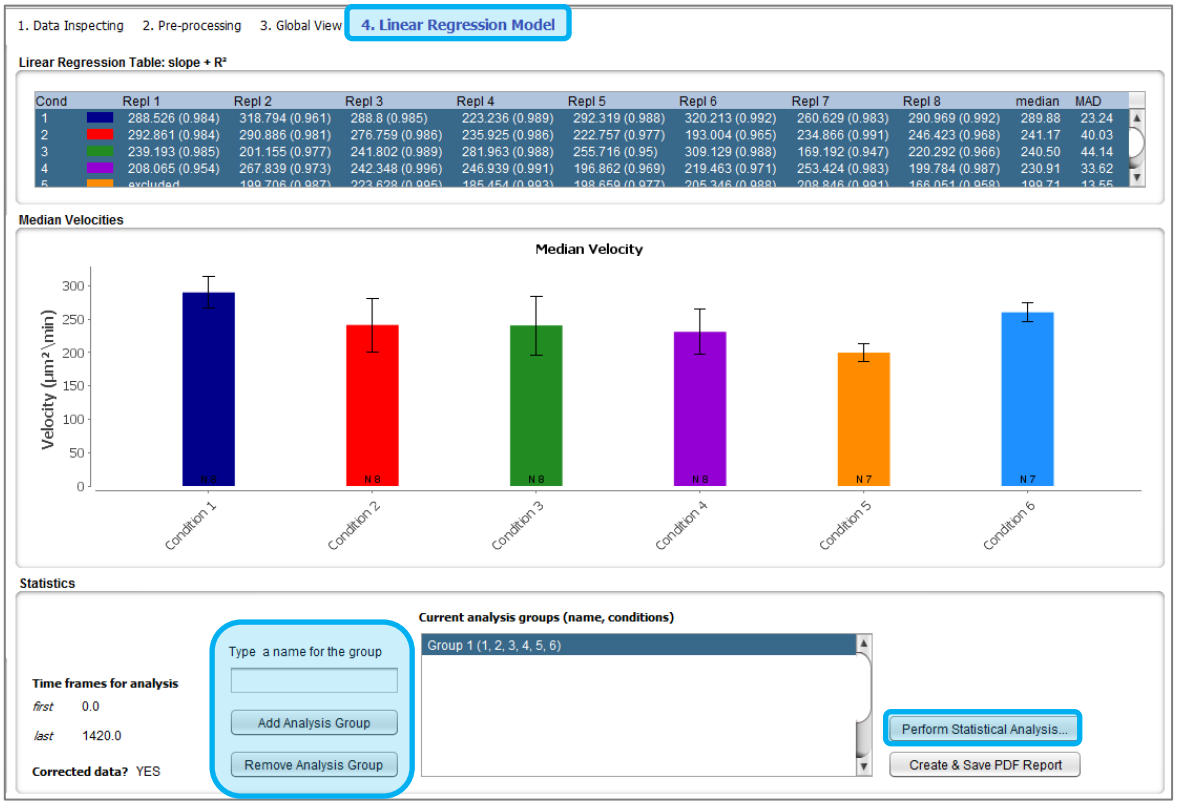


4. ***Cell migration velocity calculation (based on Linear Regression Model) and statistical comparison of biological conditions.***

   Here, **CellMissy** currently makes use of a linear regression model to extract a slope and R^2 of the area over time plot for each replicate in a biological condition. The median slope of the replicates provides then the median velocity of cell migration/wound closure for the biological condition. The bar chart in this view presents this median velocity computed for each biological condition (with standard error of the median). Again, you can also select a subset of conditions you want to include into the plot, just selecting the rows in the table.

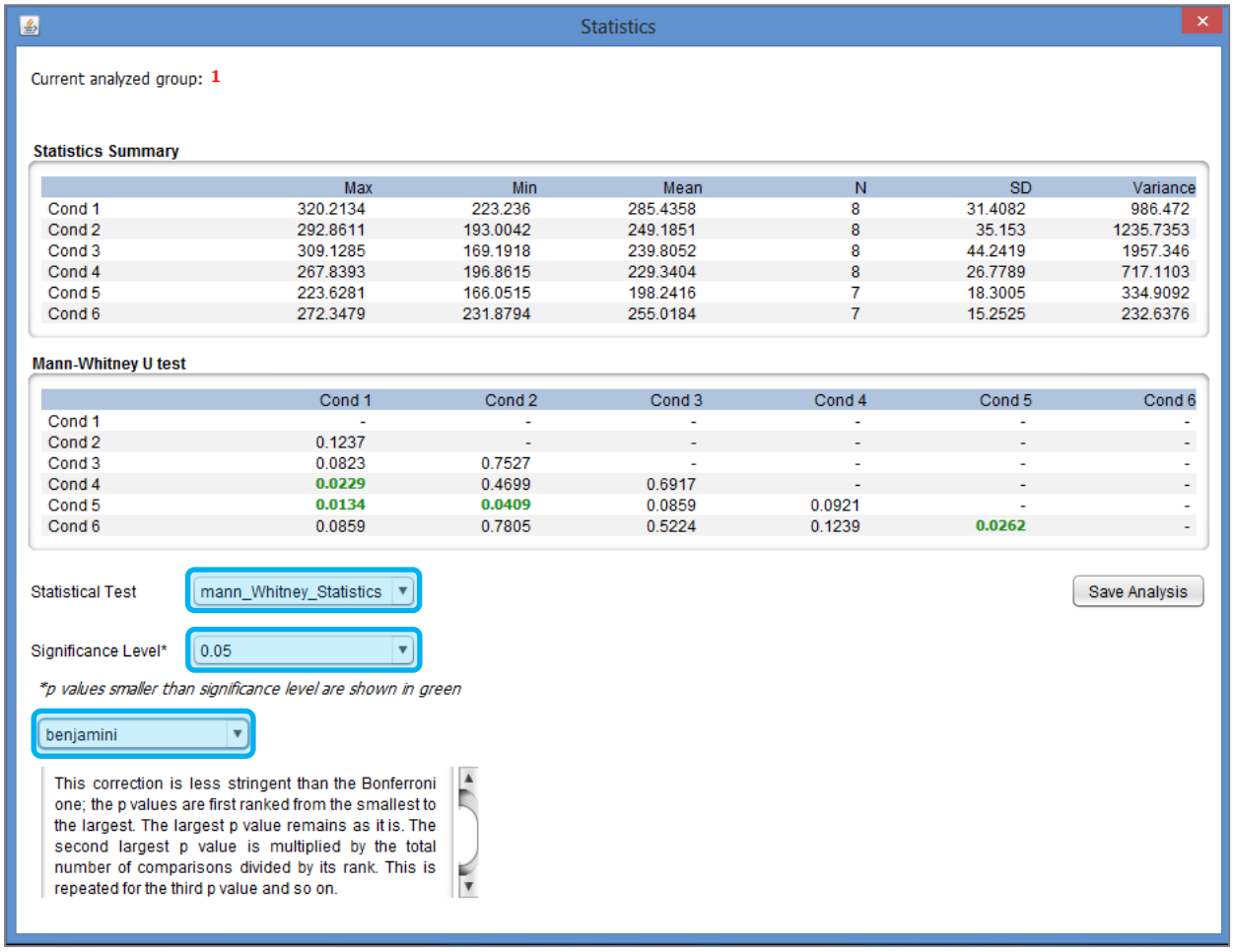
   Moreover, you can select a set of biological conditions on which you want to perform statistical analysis and provide a name for this defined group (see screenshot below).

DATA ANALYZER – Linear Regression Model



If you click the button "*Perform Statistical Analysis…*" a statistics dialog will pop up, with a summary statistics for all the conditions of the current group, and a table containing all the p-values generated by all pair-wise differences in median velocity using a Mann-Whitney U test. To correct for multiple hypothesis testing, you can either choose a Bonferroni or Benjamini-Hochberg correction. Both the statistical test and the multiple hypotheses correction methods can be easily extended with new implementation (for details, see section below 2.5). **CellMissy** provides three standard significance levels (namely 0.01, 0.05 and 0.1), and significant differences are highlighted in green in the p-values table, as you can see in the following screenshot.

DATA ANALYZER – Statistics: choosing statistical test, significance level and multiple comparison correction method



Note that every plot generated in **CellMissy** can be exported as a PNG file and can be easily customized through the *Properties* menu (just right click on a chart).

Finally, you can export a detailed *Analysis Report* in PDF that summarizes the experimental set-up, the data, results and statistics using text, tables and graphics (an example of this document is further shown in section 3.C).

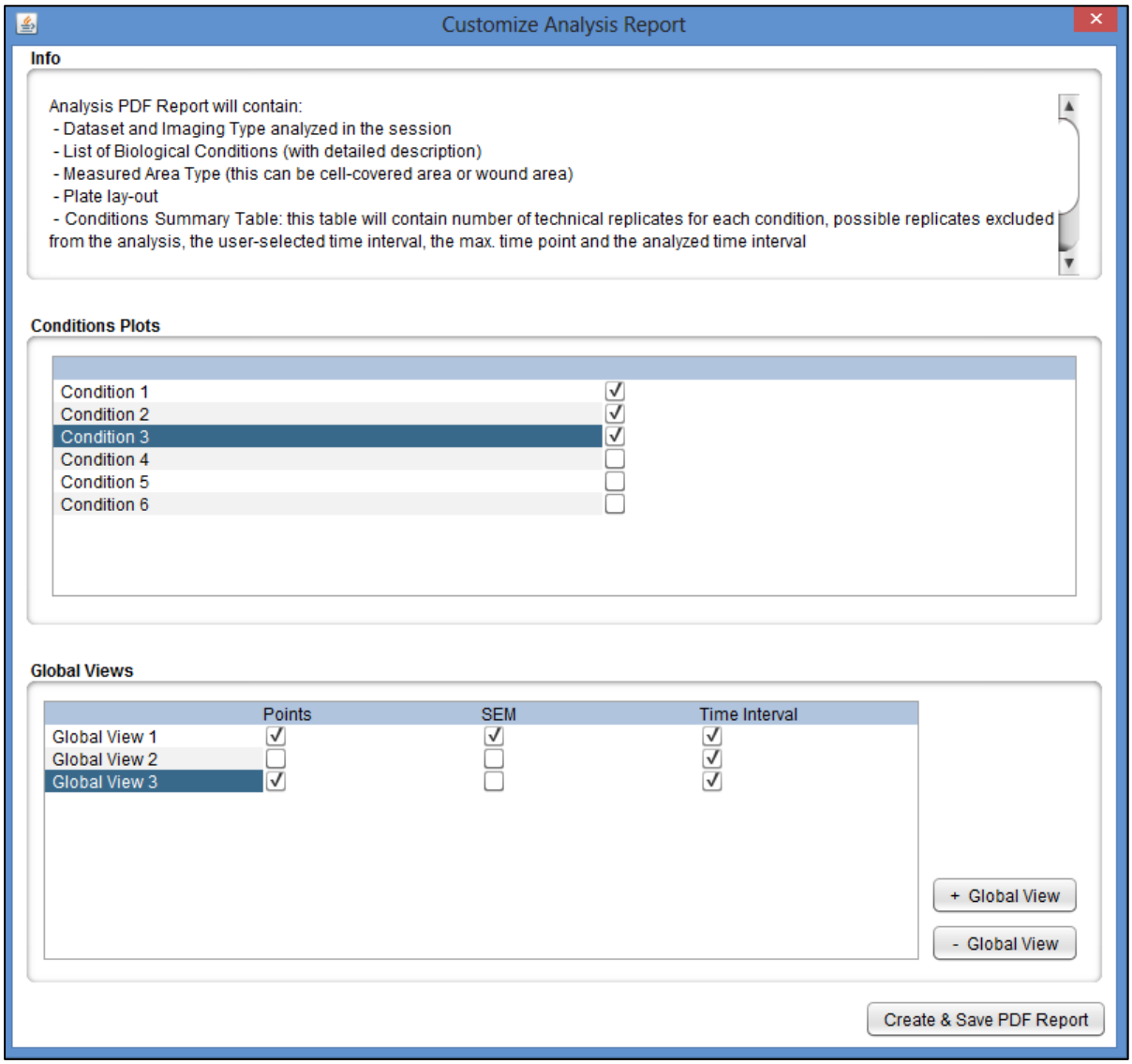
This PDF will list the following features:

- Dataset and Imaging Type analyzed in the current session
- List of Biological Conditions, along with a detailed description
- Measured Area Type (this can be cell-covered area or wound area)
- Plate lay-out reporting the experimental set-up
- Conditions Summary Table: this table will contain number of technical replicates for each condition, possible replicates excluded from the analysis, the user-selected time interval, the max. time point and the analyzed time interval

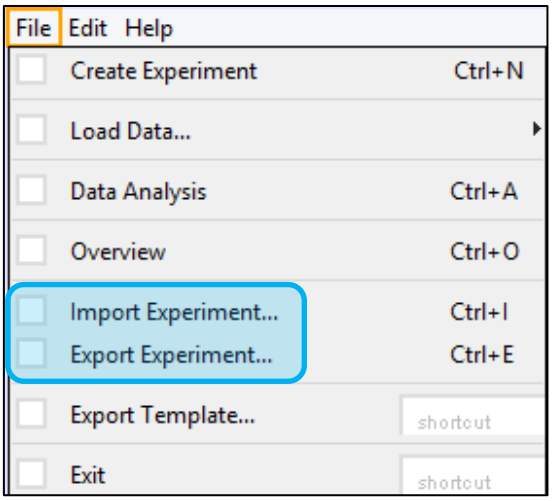
You can furthermore customize the report through a dialog (see following screenshot) that will allow you to add:

- Conditions for which you want to add area-plots
- Global Views, i.e. plots reporting the overall set of biological conditions associated to the current experiment/analysis session

DATA ANALYZER – Customizing the analysis report



2.4 DATA EXCHANGE IN CELLMISSY

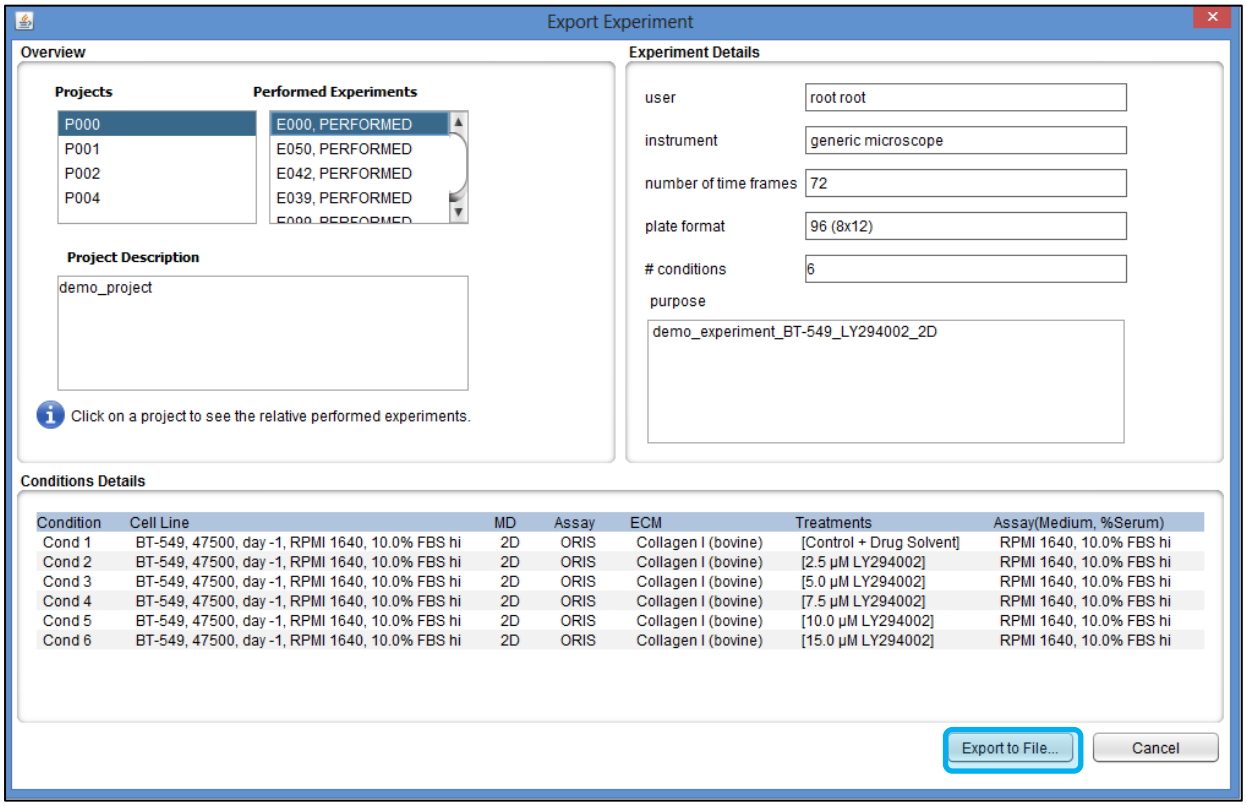


CellMissy provides a means to export an entire experiment, complete with the cell migration data, on top of the set-up settings, to an XML file. This XML file can be directly re-imported into **CellMissy**, thus enabling automatically import of an experiment to a different **CellMissy** DB, or to exchange data between different **CellMissy** environments.

2.4.1 Export an experiment

From the *File* menu of **CellMissy**, you can export an experiment to an XML file: for a project, you see all the experiments stored in the DB that are *PERFORMED*, i.e. experiments for which the area values have already been loaded. When you click on an experiment, some details are shown, along with a table reporting the biological conditions of this experiment, as shown in the next screenshot. Clicking the "*export to file…*" button will then make you choose a directory to save the file in and **CellMissy** will automatically assign a name to the file, of the type: "*experiment_EXXX_PYYY*", with *X* and *Y* to be replaced with the number of the experiment and its project, respectively. At https://cellmissy.googlecode/downloads/list you can find an example of such XML file.

DATA EXCHANGE – Exporting an experiment to XML file

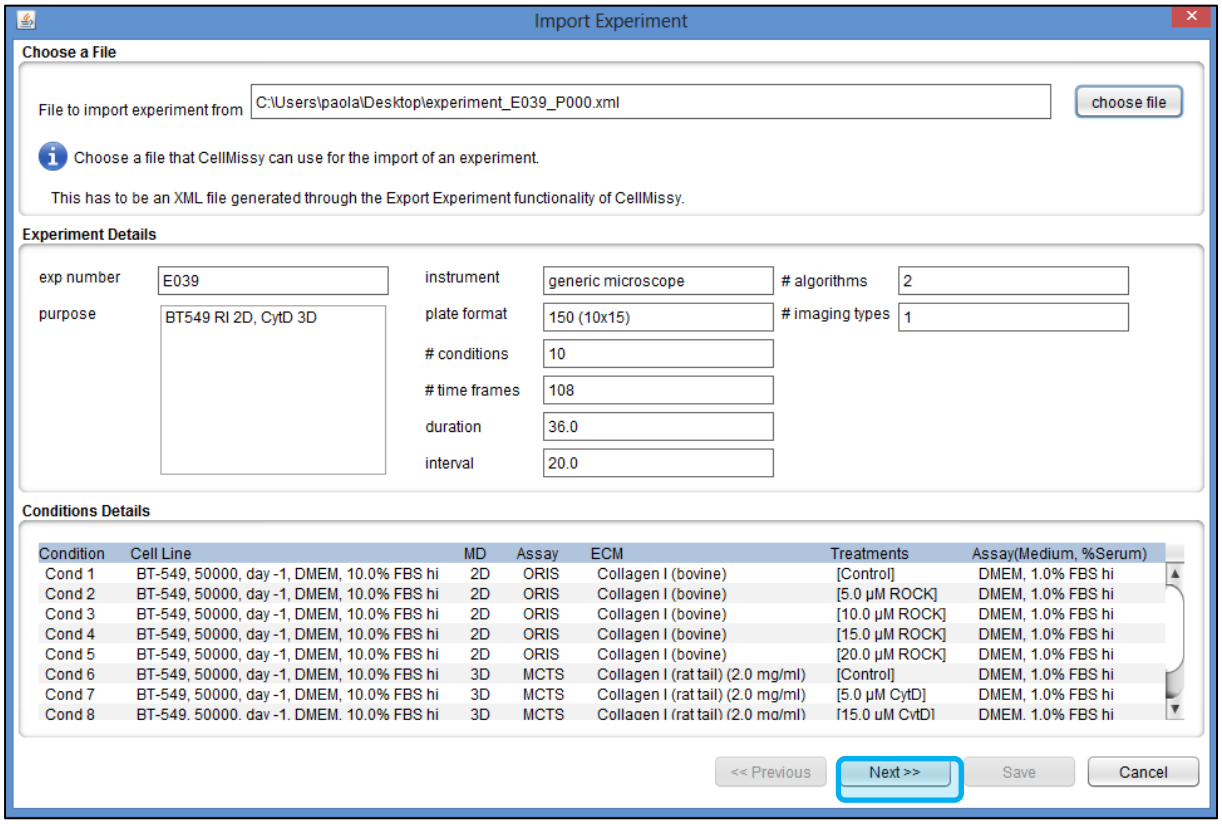


2.4.2 Import an experiment

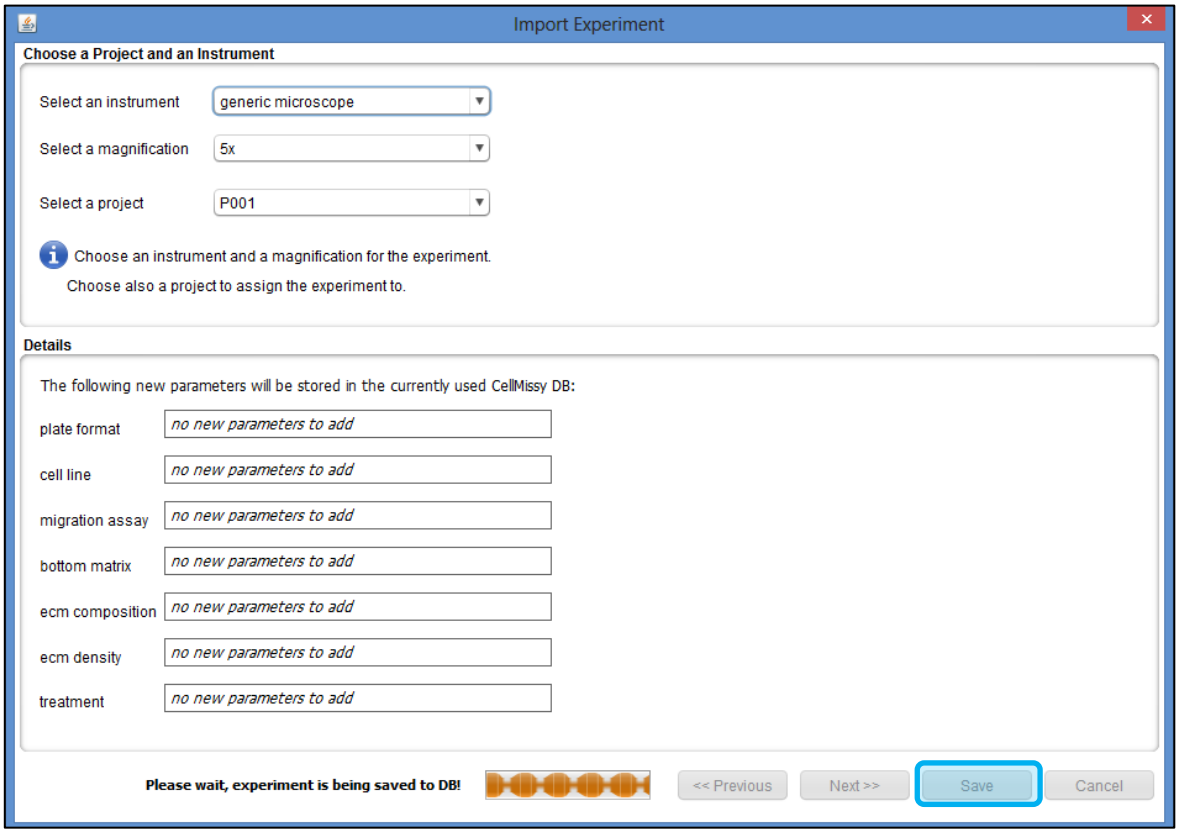
Still from the *File* menu of **CellMissy**, you can also import an experiment from an XML file; such file can be generated using the "*Export experiment*" functionality of the software illustrated in section 2.4.1. **CellMissy** shows here a dialog, where you can choose an XML file to be imported: once the file has been correctly parsed and imported, the details of the experiment are shown (including the number of algorithms and imaging types), together with a table containing all the conditions details, as shown in the following screenshot (part 1). Clicking the "*next*" button will lead you to another view (in the same dialog, see screenshot part 2), where you can choose a project and an instrument for the experiment to import. Here you also see a detailed description of what is needed in the currently **CellMissy** DB for the importing of the chosen experiment, meaning that if new parameters are associated to the imported experiment, these will be stored to the DB as well. Finally, through the "*save*" button you can proceed to save the experiment.

Please note that you can only import an experiment to a certain project if this experiment (represented by its unique number) is not present in the DB yet for that particular project! This will notably lower the possibility of errors or duplication of data.

DATA EXCHANGE – Importing an experiment from XML file – part 1



DATA EXCHANGE – Importing an experiment from XML file – part 2



2.5 EXTEND CELLMISSY: IMPLEMENT NEW ANALYSIS ALGORITHMS

The architecture of **CellMissy** is fully pluggable for analysis algorithms. This means that different distance metrics, statistical tests, kernel density estimators, outlier detection methods, and multiple hypothesis correction methods can be added by any interested developer. Such new methods can be plugged dynamically into the tool, and will subsequently be immediately available to the user from the **CellMissy** interface.

Since **CellMissy** uses the *Spring* framework for the logic layer, new implementations can be added to the software simply specifying a bean name for each new implementation, along with the fully qualified class name (i.e. including the package) in the Spring configuration XML file. Then, you just have to add the CLASS file(s) for the new implementation(s) to the "*ext*" subfolder in the **CellMissy** folder, again with the complete package structure. Finally, just run CellMissy: the new implementations will be automatically picked up and shown in the GUI.

Example – how to add a new multiple testing correction algorithm

This example will show you how to add a new multiple testing correction algorithm. By default, **CellMissy** implements two different ways of correcting for multiple hypothesis, namely the Bonferroni and the Benjamini-Hochberg correction methods; these are presented through a drop down list and you select one or the other while analyzing a certain group of biological conditions. If you want to add an extra algorithm, you need to implement the **CellMissy** *MultipleComparisonsCorrector* interface.

Let us call this new implementation (class): **MyCorrectorImpl**, and let us assume its fully qualified class name is: **com.compomics.cellmissy.cellmissytest.MyCorrectorImpl**.

Thus, in the Spring Configuration file present in the **CellMissy** folder (*mySpringXMLConfig.xml*), we will add a line containing the information about the new implementation, as shown below:

```
<!-- analysis implementations beans -->
<!-- specify a bean name for each implementation, and the fully qualified classname, i.e.: including package -->
<bean id="benjamini" class="be.ugent.maf.cellmissy.analysis.impl.BenjaminiCorrector">
</bean>

<bean id="bonferroni" class="be.ugent.maf.cellmissy.analysis.impl.BonferroniCorrector">
</bean>

<bean id="my_corrector" class="com.compomics.cellmissy.cellmissytest.MyCorrectorImpl">
</bean>
```

Next, we will put our MyCorrectorImpl.class file in the "*ext*" folder, along with the entire package structure, i.e. we will add **com.compomics.cellmissy.cellmissytest**, with cellmissytest ultimately containing the new MyCorrectorImpl.class file.

NOTE: make sure you add both a name for the new bean and the fully qualified class name for the implementation in the Spring file, and be sure as well you build the new implementation with the same JAVA version in which CellMissy has been built (by default, this is JAVA 1.7 or above).

Any other analysis algorithm can be easily extended in **CellMissy** in the same way; below is a list of the extensible feature and the correspondent interface to implement:

Outliers Detection: ***DistanceMetricOperator*** interface
Kernel Density Estimation: ***KernelDensityEstimator*** interface
Distance Metric: ***DistanceMetricOperator*** interface
Statistical Test: ***StatisticsCalculator*** interface

3. EXAMPLE DATA

A. EXAMPLE DATA TO USE WITH GENERIC INPUT FORMAT

Experiment type: scratch wound assay performed in a multiwell (12-well format) *(Tondeleir et al. PMID: 22448045 and unpublished):* 2D migration on a coating of monomeric collagen Type 1.

Biological Conditions:

Condition	Cell line	Treatment	Number of replicates
1	MEF*	None/control	3
2	MEF$^{beta\text{-}actin\text{-}/\text{-}}$	None/control	3
3	MEF	10µM Y27632**	3

**MEF: mouse embryonic fibroblast*

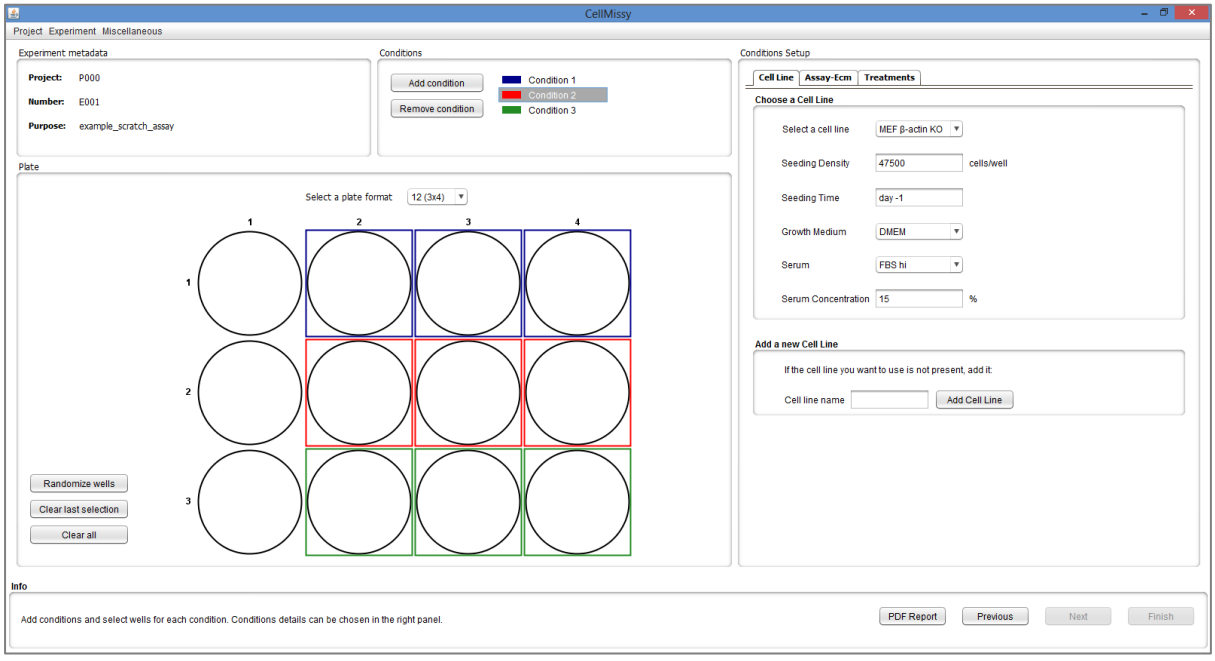
***Y27632: Rock inhibitor*

Data acquisition: 25 measurements during 12.5 hours, 30 minutes time interval

Image processing: ImageJ (NIH), manual delineation of area, output in pixel.

The following screenshot shows how the set-up of this experiment looks like.

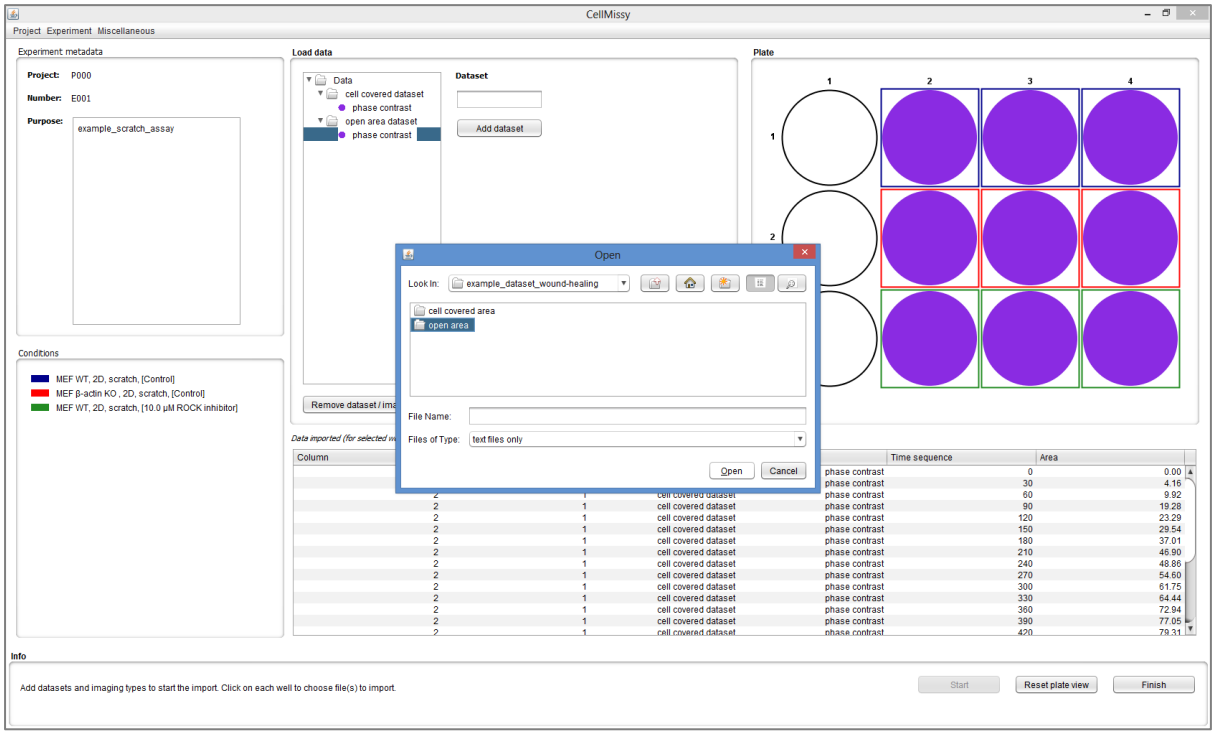
SCRATCH WOUND ASSAY - EXPERIMENTAL SETUP



At https://cellmissy.googlecode/downloads/list, you can download the "*example_dataset_scratch*" folder, where you find two sets of text files you can use for generic

import in **CellMissy**. One set of data contains the area values obtained measuring the open area in this experiment, while the other one contains the cell-covered area values in turn produced from the experiment. The next screenshot shows the data importing for this experiment.

SCRATCH WOUND ASSAY – DATA LOADING



The output files generated by **CellMissy** are also present in the same folder, namely in the "*output*" subfolder. Please note that each selection of a specific time interval produces in **CellMissy** a different analysis session, consequently generating a different PDF report. In the analysis reports folder of the scratch dataset you will find for the cell-covered dataset two different PDF reports, one generated in a session where the full time interval was taken into account, and another one generated in a session where only a subset of the time interval was considered. The open area dataset has instead been analyzed in the full time frame, providing one PDF report.

B. EXAMPLE DATA AUTOMATICALLY LOADED IN THE SOFTWARE

Experiment type: Cell exclusion zone assay performed in a 96-multiwell plate, 2D migration on coating of monomeric collagen Type I.

Biological Conditions:

Condition	Cell line	Treatment	Number of replicates
1	BT-549*	None, only drug solvent f.c. solvent: 0.2% DMSO	8
2	BT-549	2.5 µM LY249002**	8
3	BT-549	5 µM LY249002r	8
4	BT-549	7.5 µM LY249002	8
5	BT-549	10 µM LY249002	8
6	BT-549	15 µM LY249002	8

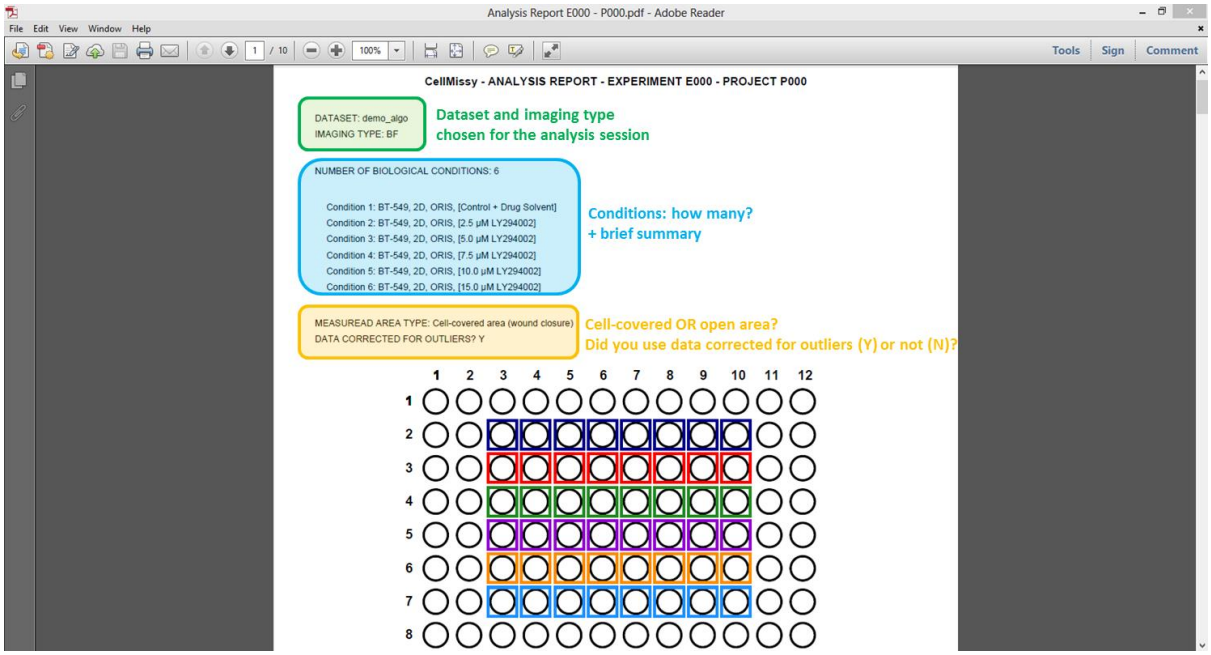
**BT549: breast cancer cell line*

***LY249002: PI3K inhibitor, drug solvent DMSO, fc in assay 0.2%*

Data acquisition: 72 measurements during 24 hours, 20 minutes time interval
Image processing: Customized automated software developed in our group (CELLMIA, Huyck et al., *in preparation*, collaboration with DciLabs) which measures the cell-covered area (in μm^2).

At https://cellmissy.googlecode/downloads/list, you can download the "*example_dataset_ORIS*" folder, where you find a SQL script (the "*example_dataset_ORIS*" file) to automatically set up and load the data for this experiment in **CellMissy**. Running this SQL script will store the area data to the database, allowing you to visualize, explore and analyze the data with the Data Analyzer module. The output file generated by **CellMissy** is also present in the same folder, namely in the "*output*" subfolder.

C. EXAMPLE OF AN ANALYSIS REPORT (ANALYSIS REPORT E000 - P000_OPEN AREA, ALSO AVAILABLE ONLINE)



CONDITIONS SUMMARY

CONDITIONS	# TECHNICAL REPLICATES	TECHNICAL REPLICATES EXCLUDED?	USER SELECTED TIME INTERVAL	MAX. TIME POINT	ANALYZED TIME INTERVAL
	8	N	(0, 71)	71	(0, 40)
	8	N	(0, 71)	71	(0, 40)
	8	N	(0, 71)	71	(0, 40)
	8	N	(0, 71)	71	(0, 40)
	7	Y [(3, 6)]	(0, 40)	71	(0, 40)
	7	Y [(3, 7)]	(0, 47)	71	(0, 40)

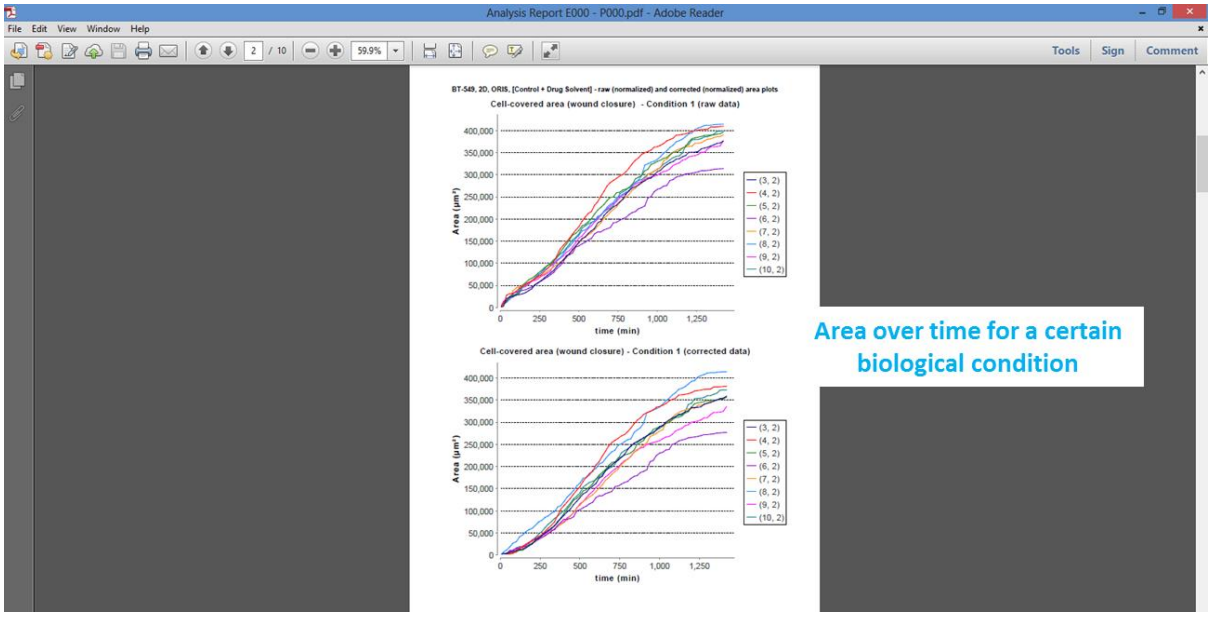
Number of replicates kept in the analysis

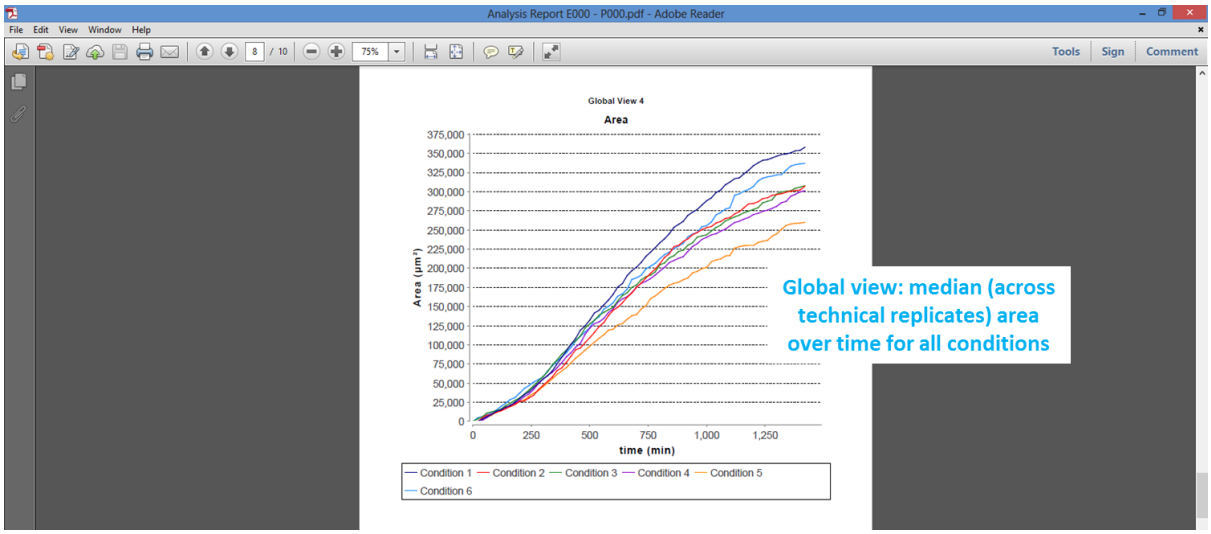
Technical replicates were excluded (Y) or not (N)?

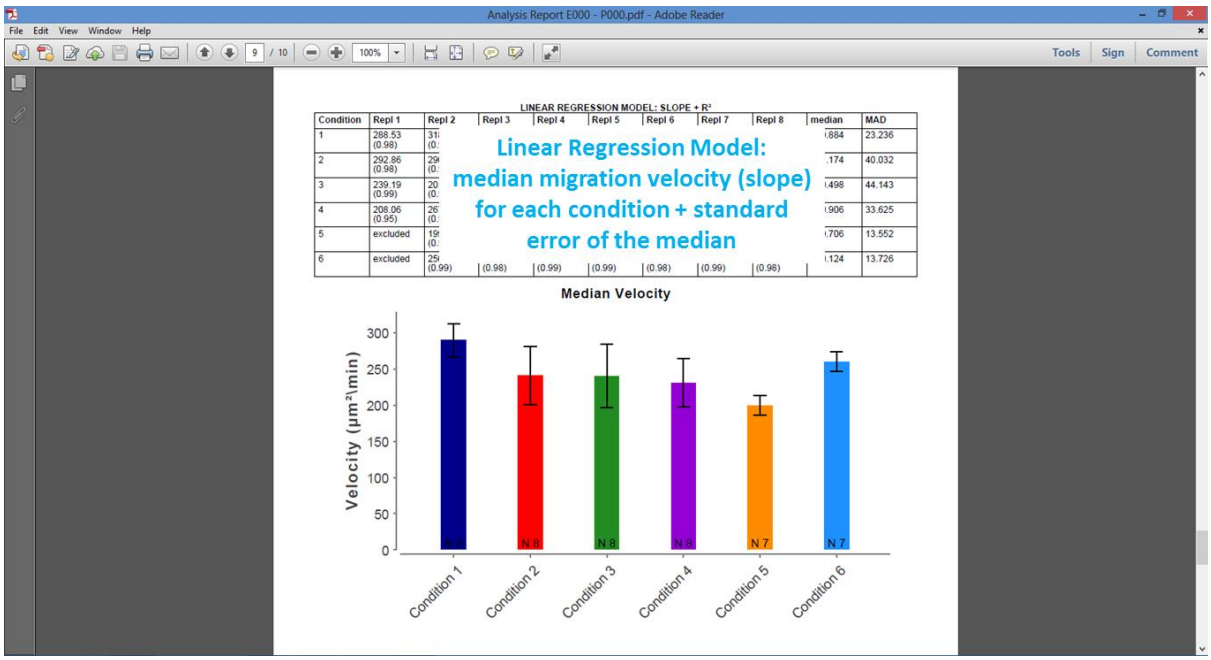
Time interval selected by the user

Suggested last time point: from this point on, all the conditions have not-null area values

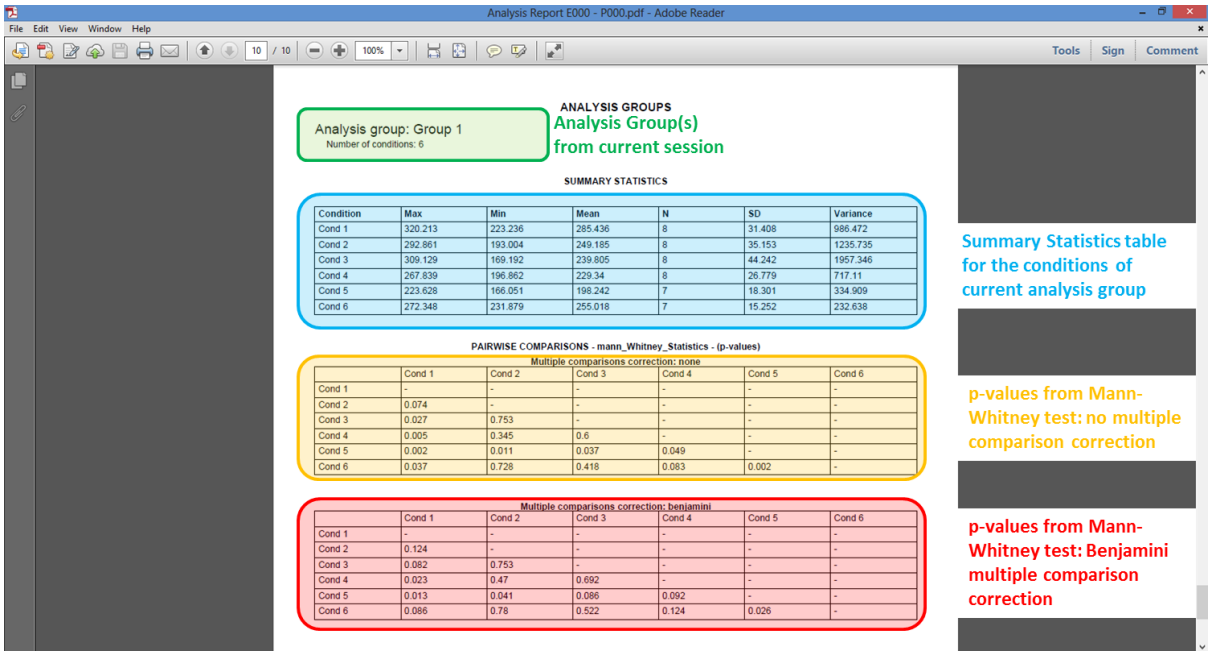
(firt, last) time points used for the analysis


Analysis Report E000 - P000.pdf - Adobe Reader
File Edit View Window Help
Tools Sign Comment
BT-549, 2D, ORIS, [Control + Drug Solvent] - raw (normalized) and corrected (normalized) area plots
Cell-covered area (wound closure) - Condition 1 (raw data)
Area (µm²)
time (min)
(3, 2)
(4, 2)
(5, 2)
(6, 2)
(7, 2)
(8, 2)
(9, 2)
(10, 2)
Cell-covered area (wound closure) - Condition 1 (corrected data)
Area over time for a certain biological condition


Analysis Report E000 - P000.pdf - Adobe Reader
File Edit View Window Help
Tools Sign Comment
Global View 4
Area
Area (µm²)
time (min)
Condition 1
Condition 2
Condition 3
Condition 4
Condition 5
Condition 6
Global view: median (across technical replicates) area over time for all conditions



Condition	Repl 1	Repl 2	Repl 3	Repl 4	Repl 5	Repl 6	Repl 7	Repl 8	median	MAD
1	288.53 (0.98)	31: (0.							.884	23.236
2	292.86 (0.98)	29 (0.							.174	40.032
3	239.19 (0.99)	20 (0.							.498	44.143
4	208.06 (0.95)	26 (0.							.906	33.625
5	excluded	19 (0.							.706	13.552
6	excluded	25 (0.99)	(0.98)	(0.99)	(0.99)	(0.98)	(0.99)	(0.98)	.124	13.726



Condition	Max	Min	Mean	N	SD	Variance
Cond 1	320.213	223.236	285.436	8	31.408	986.472
Cond 2	292.861	193.004	249.185	8	35.153	1235.735
Cond 3	309.129	169.192	239.805	8	44.242	1957.346
Cond 4	267.839	196.862	229.34	8	26.779	717.11
Cond 5	223.628	166.051	198.242	7	18.301	334.909
Cond 6	272.348	231.879	255.018	7	15.252	232.638

	Cond 1	Cond 2	Cond 3	Cond 4	Cond 5	Cond 6
Cond 1	-	-	-	-	-	-
Cond 2	0.074	-	-	-	-	-
Cond 3	0.027	0.753	-	-	-	-
Cond 4	0.005	0.345	0.6	-	-	-
Cond 5	0.002	0.011	0.037	0.049	-	-
Cond 6	0.037	0.728	0.418	0.083	0.002	-

	Cond 1	Cond 2	Cond 3	Cond 4	Cond 5	Cond 6
Cond 1	-	-	-	-	-	-
Cond 2	0.124	-	-	-	-	-
Cond 3	0.082	0.753	-	-	-	-
Cond 4	0.023	0.47	0.692	-	-	-
Cond 5	0.013	0.041	0.086	0.092	-	-
Cond 6	0.086	0.78	0.522	0.124	0.026	-