Deliverable D4.8

Funding Scheme: THEME [ICT-2007.8.0] [FET Open]

Paving the Way for Future Emerging DNA-based Technologies: Computer-Aided Design and Manufacturing of DNA libraries

Grant Agreement number: 265505

Project acronym: CADMAD

Deliverable number: D4.8

Deliverable name: Prototype DNA editor that computationally efficiently monitors the results of DNA editing (Qc etc)

Contractual Date[1] of Delivery to the CEC: M36
Actual Date of Delivery to the CEC:
Author(s)[2]: Weizmann
Participant(s)[3]: Weizmann, ETHZ (Sterling)
Work Package: WP2
Security[4]: Pub
Nature[5]: R
Version[6]: 0.0
Total number of pages: 6

1 As specified in Annex I

2 i.e. name of the person(s) responsible for the preparation of the document

3 Short name of partner(s) responsible for the deliverable

4 The Technical Annex of the project provides a list of deliverables to be submitted, with the following classification level:
Pub - Public document; No restrictions on access; may be given freely to any interested party or published openly on the web, provided the author and source are mentioned and the content is not altered.
Rest - Restricted circulation list (including Commission Project Officer). This circulation list will be designated in agreement with the source project. May not be given to persons or bodies not listed.
Int - Internal circulation within project (and Commission Project Officer). The deliverable cannot be disclosed to any third party outside the project.

5 **R (Report):** the deliverables consists in a document reporting the results of interest.
P (Prototype): the deliverable is actually consisting in a physical prototype**,** whose location and functionalities are described in the submitted document (however, the actual deliverable must be available for inspection and/or audit in the indicated place)
D (Demonstrator): the deliverable is a software program, a device or a physical set-up aimed to demonstrate a concept and described in the submitted document (however, the actual deliverable must be available for inspection and/or audit in the indicated place)
O (Other): the deliverable described in the submitted document can not be classified as one of the above (e.g. specification, tools, tests, etc.)

6 Two digits separated by a dot:
The first digit is 0 for draft, 1 for project approved document, 2 or more for further revisions (e.g. in case of non acceptance by the Commission) requiring explicit approval by the project itself;
The second digit is a number indicating minor changes to the document not requiring an explicit approval by the project.

Abstract

Development of computational and experimental tools for monitoring the accuracy and quality of liquid handling robot operation is a critical component of automation intensive biological systems. Specifically, the DNA editing system by CADMAD uses automation intensively and is therefore in need of an efficient system for automation quality control. During the past 3 years, and specifically, during the last year, we have been developing tools for monitoring the accuracy of liquid handling in CADMAD automation scripts and used them to both fix and maintain the system routinely.
Specifically, during the last year we have (1) developed computational tools for archiving, monitoring in real time and analysing the data from these routine monitoring tests and the production process itself in order to gain better understanding of the dynamics of robotic malfunction over time and identify recurring problems in a more statistically meaningful manner. Additionally, we have (2) re-written our automation module from Matlab into Python (a programming language) which enables improved tracking of DNA editing reagents in our databases, which improves our ability to monitor the system in operation.
These last year developments, together with the development of the liquid handling accuracy tests developed during the first 2 years consist of the most complete integrated QC system for liquid handling robots that we know of and are an indispensable tool for CADMAD.

Keywords[7]**:**

Automation, QC, Real-time monitoring, Liquid handling

Introduction

a. Aim / Objectives

To develop a system that efficiently monitors the accuracy of the CADMAD DNA editing system

b. State of the Art

The only QC system that we know of is a commercial kit by ‘Artel’ that measure liquid handling accuracy in a closed set of several pre-defined scripts.

c. Innovation

There are several innovations in our QC system that make it superior to any existing commercially available or other system in the field of automation:

1) Customization - Our system enables the specific testing of CADMAD (or any other) scripts for liquid handling, in contrast to existing technology which only tests a few operations defined by the commercial provider, which often to not directly test the operation that needs to be tested.
2) Computational monitoring over time – our system enables tracking and analysis of QC results over time, which often leads to insights and revision of the automation scripts.
3) Real-time monitoring – our system alerts its users of robot malfunction in real-time, which makes working with the robot more efficient, especially during the run of long scripts (some take hours).

2. Implementation

[7] Keywords that would serve as search label for information retrieval

The following actions were taken during the past year in order to achieve a fully functional DNA editing system that monitors the accuracy of its operation:

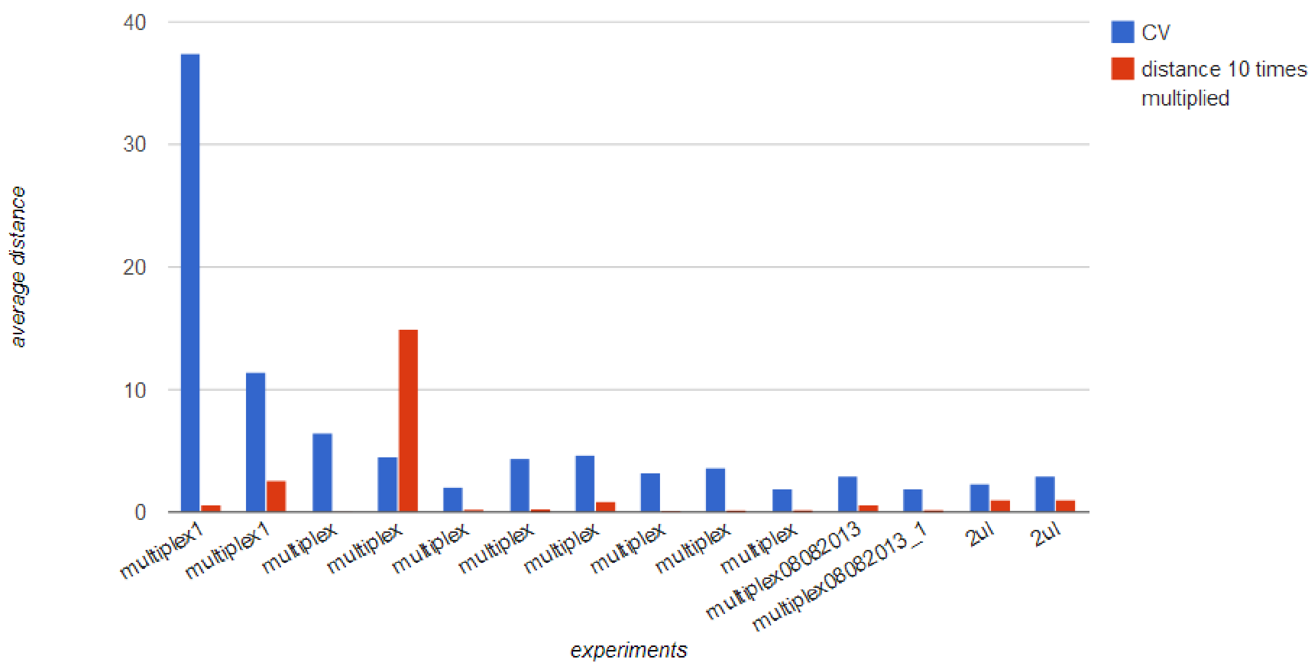
1) Development of real-time robot malfunction notification
2) Development of computational archiving and analysis capabilities for the routine QC tests
3) Rewriting of the automation module (Matlab to Python), to enable better reagent tracking during library production

3. Results

The results of the implementation above is that the DNA editing system is (1) routinely tested for the quality of its liquid handling, (2) scripts are fine-tuned routinely according to results of QC, (3) new scripts are tested with the QC system, (4) robot malfunction is reported in real-time, saving wasted human time, and (5) robot QC results are analysed over time in a more statistically significant manner, enabling us to gain perspective on it function.
As a whole, the DNA editing system is now fully controlled and monitors almost every aspect of its robotic function, as well as the quality of the DNA construction process itself.

Development of a web-based QC system for monitoring the quality and problems that occur in robot pipetting over time.

In the 24M report we showed the development of a QC system that



A report produced using our improved online robotic QC application
The above graph shows the average distance of 2ul dispense from the expected volume in red
and the CV in blue. It is are taken from our new and improved online QC system hosted at http://robotqcapp.appspot.com/.
The constant use of the system has allowed us to use the robot in high throughput robotic scripts using low volume pipetting, which is very difficult to perform manually. A report produced using our online robotic QC application. The above graph shows the average coefficient of variant values (CV) over time in different experiments in robotic scripts that use low volume dispensing of 2ul. The initial optimization and debugging phase (first 4 experiments) shows a large improvement in accuracy (7 fold decrease in CV), followed by routine monitoring in order to verify that the robot is working properly.

The report summarizes the pipetting results on the robot using a specific liquid class and a specific volume. Similarly, any report can be produced using our new online robotic QC system for all the scripts ran on the robot using any liquid class and any volume.

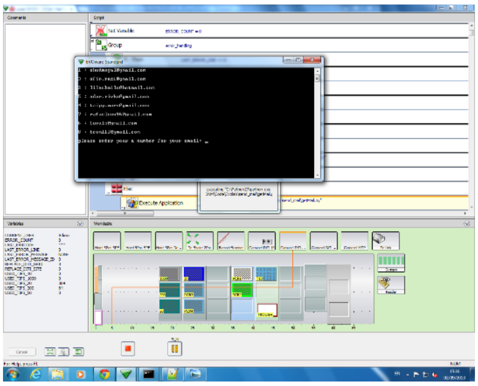
Development of a web-based system for online error alert and documentation of robot malfunction errors

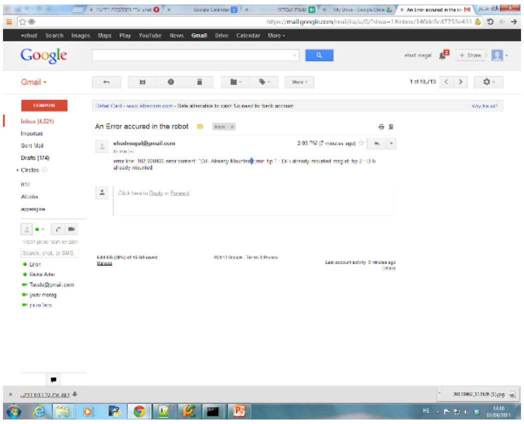
There is a real need for a real time messaging and documentation system in order to both monitor scripts running in real time and document they types of errors that occur on the robot over time.

Robot errors number increases as the number of people working with the robot and number of hours the robot works increases. For example, typical robotic script are more than 700 lines long, involving using hundreds of disposable tips. When a 96 tips plate is consumed, the robot switches the empty plate with a new plate. This process is error prone because it involves many parts of the robot moving together. These types of errors need to be monitored in real time to alert users of problems and also documented for debugging purposes. As a results of these needs, we have developed a nanline system that both alerts users in real time and documents the errors.

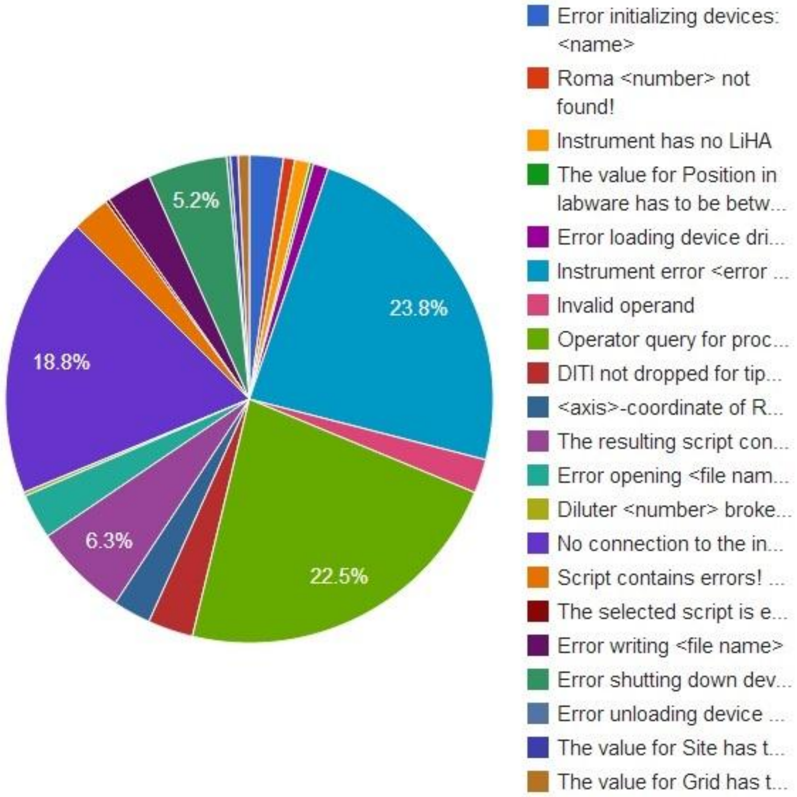
Below is a brief description of the flow of the system:

Before running of a script on the robot and email is collected from the user to a file



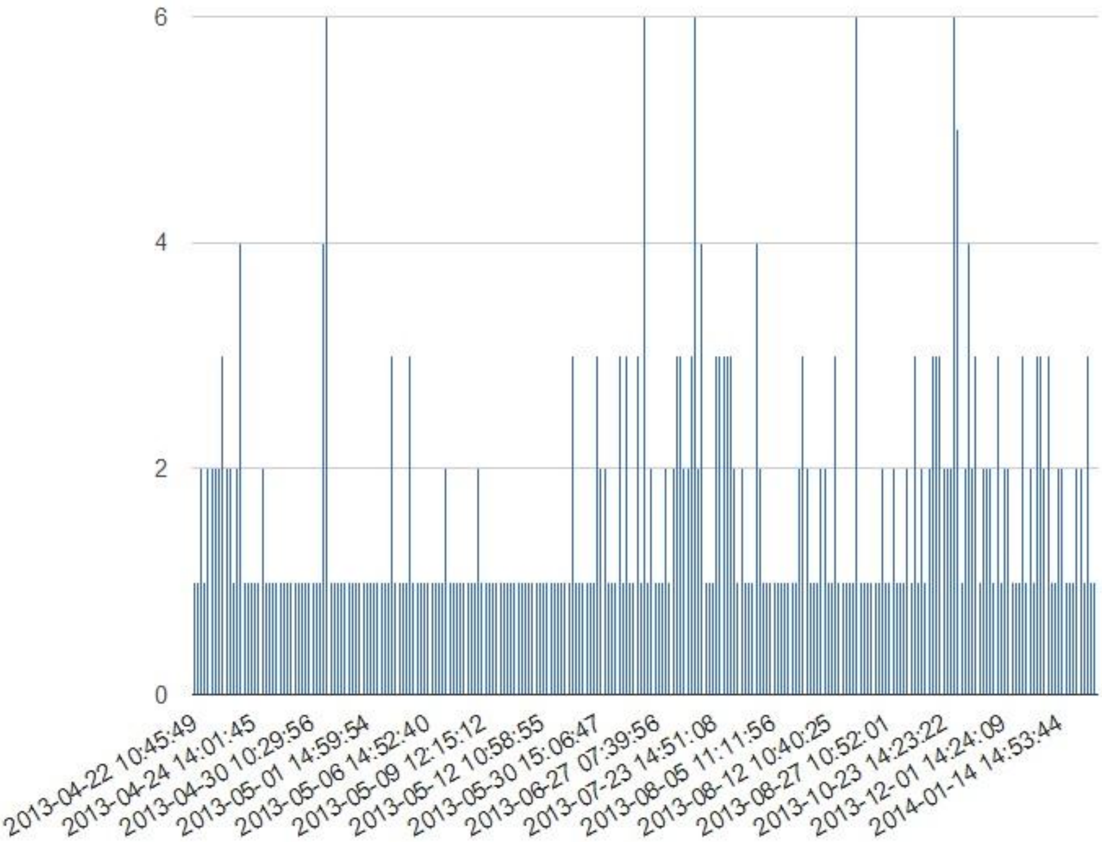


A python program is called by the evo software to send an email with the error to the user for alert.



The pie chart above shows the analysis of tracking robor malfunctions over time. 1468 individual robot errors divided in this pie chart by the type of error message.
For example errors with the message "DITI not dropped for tip<number>.Retry?" represent 3% of all errors run on the robot.
The python program also sends the error to a web server, for documentation.
For each error the server documents 4 parameters:
(1) Error line, (2) error message, (3) error code, (4) date and time. This error documentation system enables us, over time, to gain statistical knowledge about recurring errors in running the robot.

An additional analysis our QC system performs is the tracking of the number of errors that occur for each scripts that is run on the robot, as in the graph below:



The figure above shows an analysis from our QC system. Each bar on the graph (251 in total) represents a script that the robot executed. The x axis represents all the scripts (251) ordered by time (some dates indicated, script identity is not indicated here due to figure size constraints) and the y axis represents the number of errors that occurred during each script. For example: during the script run on the robot at 2013-10-30 16:42:37 the robot had 6 errors (each recorded and analysed separately).

This tool enables us to monitor error frequencies per time and script more efficiently.

4. Conclusions

The results above show that we have a fully functional, self-testing DNA editing system.