# Deliverable D5.5

**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: D5.5**

**Deliverable name: Operational prototype system constituting the complete CAD and CAM of a DNA library**

| |
|---|
| **Contractual Date[1] of Delivery to the CEC: M36** |
| **Actual Date of Delivery to the CEC:** |
| **Author(s)[2]: Sandra Meyer, Frank Edenhofer, Jean Dubern** |
| **Participant(s)[3]: UKB, UNOTT** |
| **Work Package: WP5** |
| **Security[4]: Pub** |
| **Nature[5]: R** |
| **Version[6]: 0.0** |
| Total number of pages: 11 |

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

The goal of this work package is to provide a basis for generation of an interface between computer scientists and biologists, thereby bringing together developers of the DNA programming tools and potential end users of the system. This deliverable describes the actual status of library synthesis for DNA libraries designed using DNA library designer software DNALD and processed by the novel platform technology devised in the other work packages.



## Keywords[7]:

DNA library design, validation of CADMAD system


## Introduction

### a. Aim / Objectives

CADMAD aims at replacing conventional *de novo* DNA synthesis not only by high throughput computer-aided and automated DNA processing but also by exploiting DNA reuse on a large scale. The success of CADMAD technology will critically depend on a powerful and user-friendly interface and a high level of end user compliance. Work package 5 is intended to provide a profound basis for the generation of reliable and robust interface between computer scientists, who develop the DNA processing tools, and biologists, as potential end users of the developed system. This deliverable deals with the description of the actual status of DNA library synthesis. Thus it is another step to achieve our goal of testing and challenging the various subsystems of the CADMAD system. The libraries synthesized by CADMAD technology will, in the remaining time of the project, be used to verify the breakthrough that computer-aided design and manufacturing can be effectively employed in DNA-based research and development.

### b. State of the Art

At present the direct synthesis of genes is the most efficient way to generate functional genetic constructs. For this *de novo* synthesis activated monomers (protonated deoxyribonucleoside 3'-phosphoramidites) are sequentially added to a growing chain that is linked to an insoluble substrate (Letsinger and Mahadevan, 1965; Caruthers et al., 1987). In brief, the 3' phosphorus atom of the monomer is joined to the 5' oxygen atom of the growing chain, whereas a protecting group blocks the 5' -OH group of the monomer. In the next step the generated phosphite triester is oxidized to form a phosphotriester. Finally, the protecting group on the 5'-OH is removed, so that another monomer can

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

be added to the growing chain in the next cycle. Single-stranded oligonucleotides with a length up to 100 monomers can be generated with this method. These oligonucleotides can then be enzymatically assembled to form complete synthetic genes.

In practice, several companies offer conventional gene synthesis (e.g. GenScript, Integrated DNA Technologies, Life Technologies and various others). Generally, they serve the end user with an internet-based order interface, in which the desired DNA-sequences can be uploaded by straightforward “copy and paste” action. Some companies (e.g. GenScript, Life Technologies and others) provide tools to further customize the DNA-synthesis to fulfil the needs of scientists such as codon optimization. Additionally, the DNA-sequence can be modified by removal of cryptic splice sites and RNA destabilizing sequence elements. These gene optimization steps appear to result in a maximized expression of the synthetic gene in the desired expression systems. The desired DNA-sequence is then synthesized and cloned in an output vector of choice before delivery to the end user. However, besides these obvious advantages also limitations become evident, that should be eliminated by using automated DNA processing. In this regard, the most important point is the time- and money-consumption of conventional gene synthesis. CADMAD aims at improving this point by reassembly and rearrangement of existing DNA fragments instead of conventional *de novo* DNA synthesis.

### c. Innovation

To achieve the goal of automated DNA processing a new textual DNA programming language and a new user interface as well as biochemistry and algorithms for computer-based approaches were developed throughout the course of the project. Further processing of DNA fragments via robotics and microfluidics was automated. After production of libraries, the output DNA has to be further analysed, verified and compared to conventionally synthesized DNA by applications by the end user to test and challenge the functionality and user friendliness of the CADMAD system.

## 2. Implementation

To verify the CADMAD system, the end users have drafted seven DNA libraries from various independent research fields and scientific backgrounds:

| Number of library | Short name of partner | Title of library |
|---|---|---|
| 1 | ETHZ | Combinatorial synthetic operon library |
| 2 | FMI | Sequence replacement library to identify determinants of CpG islands methylation states |

| | | |
|---|---|---|
| 3 | UNEW | DNA origami |
| 4 | UH | Glycosylation screening of the hRET-ECD |
| 5 | UKB | Derivation of Oct4 expression library to identify enhanced variants of Oct4 transcription factor |
| 6 | UNOTT | Investigating post-transcriptional regulation of *Pseudomonas aeruginosa* azurin by RsmA |
| 7 | UNOTT | Increase affinity of *Pseudomonas aeruginosa* PqsR quorum sensing signal receptor protein for N-oxide quinolones |

At present most of these libraries are either processed by the novel CADMAD platform technology devised in the other work packages or already put to test by the end users to perform a critical evaluation of the functionality of the CADMAD system.

## 3. Results

To validate the CADMAD system the following protocol is followed: First, the libraries are specified and designed using the recently developed DNALD software, thereby additionally defining sub-libraries that are synthesized by conventional DNA synthesis. In the remaining time of the project these fragments will be used to compare the functionality of the DNA libraries generated by the CADMAD system to the ones generated by commercial conventional DNA synthesis. Additionally, some of the commercially synthesized fragments are sent to WEIZ to be used as a basis for the generation of the complete DNA-libraries by CADMAD technology. After generation of the libraries, they are sent to the individual partners for further validation and downstream processing.

The individual libraries are in different stages of their completion: two of the DNA libraries have been already delivered to the respective partners (UNOTT: “Investigating post-transcriptional regulation of *Pseudomonas aeruginosa* azurin by RsmA” and FMI: “Sequence replacement library to identify determinants of CpG islands methylation states”), and their further validation and test of functionality is in progress. The synthesized DNA fragments needed for the generation of two other libraries have been already sent to WEIZ and the synthesis of the respective libraries is ongoing (ETHZ: “Combinatorial synthetic operon library” and UH: “Glycosylation screening of hRET-ECD”). ETHZ’s library will be delivered shortly. The last three libraries have been designed, however, production of these libraries has not been started yet (UNEW: “DNA origami”, UKB: “Derivation of Oct4 expression library to identify enhanced variants of Oct4 transcription factor” and UNOTT: “Increase affinity of *Pseudomonas aeruginosa* PqsR quorum sensing signal receptor protein for N-oxide quinolones”). The actual statuses of the individual DNA libraries are summarized in the table below:

| Number of library | Name of partner and short title of library | Specification of library? | Oligos sent to WEIZ? | Library in construction? | Library delivered? | Library tested/validated? |
|---|---|---|---|---|---|---|
| 1 | ETHZ: Combinatorial synthetic operon library | yes | yes | yes | yes (at time of review meeting) | no |
| 2 | FMI: Sequence replacement library | yes | yes | completed | yes | in progress |
| 3 | UNEW: DNA origami | in progress | no | no | no | no |
| 4 | UH: Glycosylation screening | yes | yes | yes | no | no |
| 5 | UKB: Oct4 expression library | yes | no | no | no | no |
| 6 | UNOTT: *Pseudomonas aeruginosa* azurin | yes | yes | completed | yes | in progress |
| 7 | UNOTT: *Pseudomonas aeruginosa PqsR* | yes | no | no | no | no |

Next, the properties and the process of library generation and validation will be described for the delivered library 6 (UNOTT: “Investigating post-transcriptional regulation of *Pseudomonas aeruginosa* azurin by RsmA”). The aim of this designed library was to verify that RsmA binds to the azurin mRNA coding region and stabilises it, as well as to investigate the contributions of each of the three potential RsmA binding sites present in this transcript for stabilisation. Since the last reporting period, an azurin gene library was specified (UNOTT and UNEW) which for each of the three potential RsmA binding sites had either the wild type sequence (designated P2, P4 and P6) or one of two different variations (designated P2a1, P2a2, P4a1, P4a2, P6a1 and P6a2, respectively) giving a total library size of 27 variants (including the wild type). In the variations the AGGA motifs and associated stem-loops forming potential RsmA binding sites were removed while preserving the translated sequence. All 27

variants were further designed to incorporate a C-terminal 6-histidine marker in the coding sequence to enable detection by immunoblotting and the use other immunological assays. The DNALD file used to define the library is shown below:

```
library azurinLibrary {

inputs {
      azurin_PCR_product := '
AAGGT CCATGG
ATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCCGAGTGCTCGGTG
GACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGCAAGCAGTTCACC
GTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGCACCGCCGCCGAC
ATGCAGGGCGTGGTCACCGACGGCATGGCTTCCGGCCTGGACAAGGATTACCTGAAGCCCGACGACAGCCGTGTC
ATCGCCCACACCAAGCTGATCGGCTCGGGCGAGAAGGACTCGGTGACCTTCGACGTCTCCAAGCTGAAGGAAGGC
GAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACCCTGAAGTGA
GAATTC ACCTT
'}

      // wildtype with stop codon
      azurin     := azurin_PCR_product[12:458] is '
ATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCCGAGTGCTCGGTG
GACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGCAAGCAGTTCACC
GTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGCACCGCCGCCGAC
ATGCAGGGCGTGGTCACCGACGGCATGGCTTCCGGCCTGGACAAGGATTACCTGAAGCCCGACGACAGCCGTGTC
ATCGCCCACACCAAGCTGATCGGCTCGGGCGAGAAGGACTCGGTGACCTTCGACGTCTCCAAGCTGAAGGAAGGC
GAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACCCTGAAGTGA'

      // wildtype without stop codon
      // split in Parts such that no codons in 5'->3' Frame 1 are truncated
      Part_1 := azurin[1:258] is'
ATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCCGAGTGCTCGGTG
GACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGCAAGCAGTTCACC
GTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGCACCGCCGCCGAC
ATGCAGGGCGTGGTCACCGACGGCATGGCTTCC'
      Part_2 := azurin[259:273]   is 'GGCCTGGACAAGGAT'
      Part_3 := azurin[274:327]   is
'TACCTGAAGCCCGACGACAGCCGTGTCATCGCCCACACCAAGCTGATCGGCTCG'
      Part_4 := azurin[328:342]   is 'GGCGAGAAGGACTCG'
      Part_5 := azurin[343:360]   is 'GTGACCTTCGACGTCTCC'
      Part_6 := azurin[361:375]   is 'AAGCTGAAGGAAGGC'
      Part_7 := azurin[376:444]   is
'GAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACCCTGAAG' //
minus TGA stop_codon

      // Parts 1, 3, 5, 7 are to be conserved.
      // Parts 2, 4, 6 are to be altered by removing RsmA binding sites with
      AGGA and stem-loop structures.

      // 2/52 variants
      Part_2_altered_1      := 'GGCCTGGACAAAGAC'
      Part_2_altered_2      := 'GGATTAGACAAAGAC'
      Part_2_altered        :=  Part_2_altered_1 + Part_2_altered_2
```

```
    // 2/41 variants
    Part_4_altered_1      := 'GGCGAGAAAGACAGC'
    Part_4_altered_2      := 'GGAGAGAAAGATAGC'
    Part_4_altered        := Part_4_altered_1 + Part_4_altered_2

    // 2/70 variants
    Part_6_altered_1      := 'AAGCTGAAAGAGGGC'
    Part_6_altered_2      := 'AAGTTAAAAGAGGGG'
    Part_6_altered        := Part_6_altered_1 + Part_6_altered_2

    // control, reconstituted from new Part indices
    azurin_unaltered := Part_1 Part_2 Part_3 Part_4 Part_5 Part_6 Part_7

    // targets with 'AGGA' and stem-loops in non-wt Parts 2, 4 and 6
    removed
    azurin_altered    := Part_1 (Part_2 + Part_2_altered)
                         Part_3 (Part_4 + Part_4_altered)
                         Part_5 (Part_6 + Part_6_altered)
                         Part_7

    NcoI              := 'CCATGG'
    _5_prime_end      := 'AAGGT' NcoI is azurin_PCR_product[1:11]
    His_tag           := 'CAT CAC CAT CAC CAT CAC'
    stop_codon        := 'TGA' is azurin_PCR_product[456:458] is
                      azurin[445:447]
    EcoRI             := 'GAATTC'
    _3_prime_end      := His_tag stop_codon EcoRI 'ACCTT'

outputs {
    // 27 His-tagged variants with cloning sites
    azurin_library    := _5_prime_end (azurin_unaltered + azurin_altered)
                      _3_prime_end

// // 4 His-tagged variants with cloning sites
// azurin_P2a1_P4a1_P6a2    := _5_prime_end Part_1 Part_2_altered_1 Part_3
Part_4_altered_1 Part_5 Part_6_altered_2 Part_7 _3_prime_end is
'AAGGTCCATGGATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCC
GAGTGCTCGGTGGACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGC
AAGCAGTTCACCGTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGC
ACCGCCGCCGACATGCAGGGCGTGGTCACCGACGGCATGGCTTCCGGCCTGGACAAAGACTACCTGAAGCCCGAC
GACAGCCGTGTCATCGCCCACACCAAGCTGATCGGCTCGGGCGAGAAAGACAGCGTGACCTTCGACGTCTCCAAG
TTAAAAGAGGGGGAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACC
CTGAAGCATCACCATCACCATCACTGAGAATTCACCTT'
// azurin_P2_P4a2_P6a1      := _5_prime_end Part_1 Part_2 Part_3
Part_4_altered_2 Part_5 Part_6_altered_1 Part_7 _3_prime_end is
'AAGGTCCATGGATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCC
GAGTGCTCGGTGGACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGC
AAGCAGTTCACCGTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGC
ACCGCCGCCGACATGCAGGGCGTGGTCACCGACGGCATGGCTTCCGGCCTGGACAAGGATTACCTGAAGCCCGAC
GACAGCCGTGTCATCGCCCACACCAAGCTGATCGGCTCGGGAGAGAAAGATAGCGTGACCTTCGACGTCTCCAAG
CTGAAAGAGGGCGAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACC
CTGAAGCATCACCATCACCATCACTGAGAATTCACCTT'
// azurin_P2a1_P4_P6a2      := _5_prime_end Part_1 Part_2_altered_1 Part_3
Part_4 Part_5 Part_6_altered_2 Part_7 _3_prime_end is
'AAGGTCCATGGATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCC
```

```
GAGTGCTCGGTGGACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGC
AAGCAGTTCACCGTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGC
ACCGCCGCCGACATGCAGGGCGTGGTCACCGACGGCATGGCTTCCGGCCTGGACAAAGACTACCTGAAGCCCGAC
GACAGCCGTGTCATCGCCCACACCAAGCTGATCGGCTCGGGCGAGAAGGACTCGGTGACCTTCGACGTCTCCAAG
TTAAAAGAGGGGGAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACC
CTGAAGCATCACCATCACCATCACTGAGAATTCACCTT'
// azurin_P2a2_P4a1_P6      := _5_prime_end Part_1 Part_2_altered_2 Part_3
Part_4_altered_1 Part_5 Part_6 Part_7 _3_prime_end is
'AAGGTCCATGGATGCTACGTAAACTCGCTGCGGTATCCCTGCTGTCCCTGCTCAGTGCGCCACTGCTGGCTGCC
GAGTGCTCGGTGGACATCCAGGGTAACGACCAGATGCAGTTCAACACCAATGCCATCACCGTCGACAAGAGCTGC
AAGCAGTTCACCGTCAACCTGTCCCACCCCGGCAACCTGCCGAAGAACGTCATGGGCCACAACTGGGTACTGAGC
ACCGCCGCCGACATGCAGGGCGTGGTCACCGACGGCATGGCTTCCGGATTAGACAAAGACTACCTGAAGCCCGAC
GACAGCCGTGTCATCGCCCACACCAAGCTGATCGGCTCGGGCGAGAAAGACAGCGTGACCTTCGACGTCTCCAAG
CTGAAGGAAGGCGAGCAGTACATGTTCTTCTGCACCTTCCCGGGCCACTCCGCGCTGATGAAGGGCACCCTGACC
CTGAAGCATCACCATCACCATCACTGAGAATTCACCTT'
} // outputs

} // library
```

From this DNA library some output fragments were chosen which should be ordered by conventional DNA synthesis. These fragments will serve as a sub-library used to compare with the fragments generated by CADMAD technology. Four initial 477bp DNA-fragments were ordered at Integrated DNA Technologies (IDT), namely variants: azurin_P2a1_P4a1_P6a2, azurin_P2_P4a2_P6a1, azurin_P2a1_P4_P6a2 and azurin_P2a2_P4a1_P6. Additionally, the input fragment azurin_PCR_product was generated by PCR and consists of WT azurin sequence flanked by His-tag and restriction enzyme sites EcoRI and NcoI. All five fragments were individually cloned into pHERD30T vector and sequenced to prove that they were error-free. Afterwards, the cloned fragments were shipped to Weizmann Institute to be used as templates for the synthesis of the rest of the CADMAD library consisting of 22 individually derived combination variants. All these 22 remaining DNA variants were generated by WEIZ and shipped back to UNOTT. The linear DNA variants were individually arrayed on a microtiter plate. Subsequently, all variants were individually cloned into pHERD30T vector by UNOTT and quality controlled afterwards by sequencing. Herewith, eight different single nucleotide exchanges could be detected in five out of the 22 generated clones. In total 30 individual clones had to be sequenced to obtain a complete set of 22 clones with the correct, intended sequences, giving an effective misincorporation rate of approximately 6 x $10^{-4}$.

However, since in all cases error-free clones could be obtained further validation and downstream processing can be performed with all generated azurin variants. For this azurin wildtype and variants genes were cloned in vector pHERD30T, which is a shuttle expression vector that can replicate in both *E.coli* and *Pseudomonas* (Qiu et al., 2008). Expression of the azurin wild type and variant genes cloned in this vector is under the control of the arabinose-inducible *araBAD* promoter ($P_{BAD}$). The pHERD30T-based constructs were introduced into a *P. aeruginosa* PASK10L*azu*, an azurin-deficient strain (in-frame *azu* deletion mutant) of strain PAO1-L in which RsmA production has been engineered

to be controlled by varying the concentrations of IPTG in the medium (**Fig. 1**). Western blotting using anti-his antibodies of fractions obtained after growing PASKL10L*azu* carrying the pHERD30T::*azu*-$H_6$ construct encoding wild type azurin in the presence of arabinose showed that under the conditions used his-tagged azurin was mostly present in the periplasm rather than released in the medium (**Fig. 1**). However, quantification of his-tagged azurin requires a large volume of medium and a complex purification method. Thus, two alternative strategies are currently being assessed for the accurate quantification of azurin: 1) by the use of an anti-azurin antibody (produced by UNOTT) to directly detect the protein instead of via an added hexahistidine tag, and 2) by an entirely novel approach of "sandwich" hybridisation assays in which the high binding affinity between p53 and azurin (Yamada *et al*. 2002) will be exploited. The cellular tumour suppressor protein p53 has been extensively studied in cancer research and many specific detection kits for this protein are commercially available.

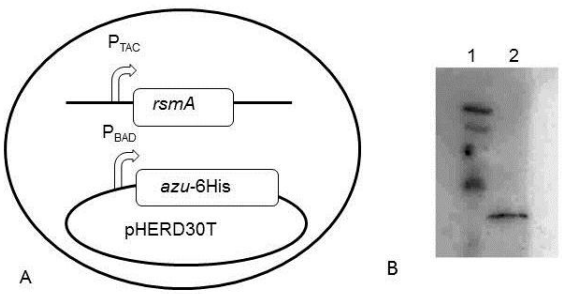


**Figure 1. A.** Schematic presentation of the *P. aeruginosa* screening strain PASK10L*azu*(pHERD*azu*). In this strain the *rsmA* gene on the chromosome is under control of the IPTG-inducible $P_{TAC}$ promoter while his-tagged azurin variant genes cloned in the shuttle vector pHERD30T are under control of the arabinose-inducible $P_{BAD}$ promoter. **B.** Western blotting carried out with anti-his antibodies. **Lane 1**. His-tagged protein ladder. **Lane 2**. Periplasmic fraction of PASK10L*azu* (pHERD30T*azu*-wt) grown in the presence of 1% arabinose.

## 4. Conclusions

A variety of DNA libraries from different biological backgrounds have been designed using recently developed DNALD software in the course of this project. The library products will be used for various scientific applications. At present most of these libraries are processed by the novel CADMAD platform technology devised in the other work packages. Two libraries have already been produced and are at present tested and validated by the end users to perform a critical evaluation of the CADMAD system. For UNOTT's library "*Pseudomonas aeruginosa* azurin" it could already been shown, that the novel CADMAD system is able to produce cloneable DNA-fragments with a low error rate. The functionality of the generated fragments in their biological context is evaluated at the moment.

## 5. References

Caruthers MH, Barone AD, Beaucage SL, Dodds DR, Fisher EF, McBride LJ, Matteucci M, Stabinsky Z, Tang JY (1987) Chemical synthesis of deoxyoligonucleotides by the phosphoramidite method. Methods Enzymol 154: 287-313

Letsinger RL, Mahadevan V (1965) Oligonucleotde synthesis on a polymer support. J Am Chem Soc 87: 3526-3527

Qiu D, Damron FH, Mima T, Schweizer HP and Yu HD (2008). $P_{BAD}$-based shuttle vectors for functional analysis of toxic and highly regulated genes in *Pseudomonas* and *Burkholderia* spp. and other bacteria. Appl. Environ. Microbiol. 74:7422–7426.

## 6. Abbreviations

*List all abbreviations used in the document arranged alphabetically.*

| | |
|---|---|
| DNA | deoxyribonucleic acid |
| DNALD | DNA library designer |
| e.g. | for example (*exempli gratia*) |
| et. al. | and others (*et alii*) |
| ETHZ | Eidgenoessische Technische Hochschule Zuerich |
| FMI | Friedrich Miescher Institut |
| IDT | Integrated DNA Technologies |
| PCR | polymerase chain reaction |
| UH | University of Helsinki |
| UKB | Universitaetsklinikum Bonn |
| UNEW | University of Newcastle |
| UNOTT | University of Nottingham |
| WEIZ | Weizmann Institute of Science |
| WT | wildtype |