Cre-ACEMBLER Synopsis

Christian Becke & Imre Berger, 2012

All ACEMBL multiexpression systems (currently MultiBac, MultiMam and MultiColi) rely on *in vitro* Cre-loxP recombination of plasmids to assemble multiprotein expression vectors. A theoretically unlimited number of plasmids can be fused in this way, currently the systems (pragmatically) foresee up to three Donor plasmids to be recombined with a single Acceptor plasmid to yield the product plasmid(s) which is the multigene expression construct(s) of choice. Each educt plasmid (Donors and Acceptor) contains a unique resistance marker to allow selection of productive fusions by multiple resistance marker challenge. This approach is made possible by the conditional origin of replication ($ori^{R6K\gamma}$) present on all Donors which requires their fusion to an Acceptor to allow for propagation in a regular bacterial host strain which does not recognize the conditional origin. The product plasmid is *quasi* barcoded by the resistance markers present on the Acceptor and Donor plasmids used in the Cre-LoxP fusion reaction [1,2,3].

It is advisable to check the recombined plasmids by restriction mapping prior to their usage for expression experiments. This is essential in the case of the MultiBac system, where the mutigene fusion plasmid needs to be integrated into the MultiBac baculoviral genome by means of Tn7 transposition [3]. The Acceptor contains the two DNA sequences required for the Tn7 transposition step (Tn7L, Tn7R) for integration into the MultiBac genome. These two sequences must each be present in a copy number of one only on the multigene fusion. Fortuitous integration of more than one Acceptor plasmids would entail the presence of several Tn7L respectively Tn7R sites, leading to non-sense integration events and concomitant failure to express the desired gene products [2].

Restriction mapping requires exact knowledge of the DNA sequence of the fusion plasmid. Since *in vitro* recombination of a mixture of plasmids occurs stochastically, all possible sequences of the recombination products have to be generated in order to predict the possible restriction patterns. The number of possible permutations Pn for a given number of plasmids n is given by the formula for circular permutations: $P_n = (n - 1)!$ [4]. For the recombination of three donor plasmids with one acceptor plasmid ($n = 4$), there are $P_4 = 6$ different possibilities for the recombination products, neglecting the possibility of fusion products containing multiple copies

of educt plasmids. To assist in generating all possible sequence permutations for designing and analysing restriction digestions of recombined plasmids, the software Cre-ACEMBLER was created [5].

Cre-ACEMBLER displays sequence data in an application window, showing the sequence as plain text. Simple manipulations can be done using cut, copy and paste functions. Sequence data can be read from and written to files in various formats, including FASTA and GenBank.

To perform *in silico* Cre recombinations, all educt plasmid sequences have to be opened in Cre-ACEMBLER. Activating the "Cre" button starts an assistant dialogue guiding through the recombination in 3 steps:

(1) Acceptor plasmid sequence is selected among all open sequences
(2) Donor plasmid sequences are selected
(3) Adjustment of the desired copy numbers of each individual plasmid

Each possible product sequence is then generated and displayed in a new window. Product sequences can then be saved to files and analysed using other software, e.g. ApE [10] or Vector NTI [11].

Main requisites to be fulfilled by Cre-ACEMBLER were (**1**) ease of use, (**2**) compatibility with a broad range of operating systems and (**3**) interoperability with other software. No central processing unit (CPU)-intensive work is done by Cre-ACEMBLER, thus an interpreted programming language could be chosen without risking performance limitations. Therefore, Cre-ACEMBLER was developed in Python [6], using the Python bindings of GTK+ [7,8] for the graphical user interface, and the Biopython [9] library for sequence data manipulations. Using Python and GTK+ allows Cre-ACEMBLER to run on Windows, Linux and MacOS operating systems, and possibly others. The Biopython library allows reading and writing sequence data in various file formats, providing good interoperability with other software.

It is of advantage for the *in silico* Cre recombination if all educt sequences contain the *loxP* sequence in the same orientation, and if the linear representation of each input sequence starts with the *loxP* site. Therefore, input sequences are normalised prior to recombination, by generating the reverse-complement of input sequences as required, and by linearising all

sequences immediately 5' of the *loxP* site. All input sequences are then indexed numerically, making sure that identical input sequences get the same index. Lists representing all possible permutations of the order of the indices are computed, and solutions which are redundant if considering circular arrangement are eliminated, thus yielding index lists representing all unique circular permutations. Fusion plasmid sequences are then generated from these index lists by appending the normalized DNA sequences corresponding to the indices, in the order given in these lists.

A challenge arising from the linear representation of circular sequences is to identify permutations which are redundant if circular arrangement is considered. In order to make the lists representing different circular solutions comparable, a linearisation algorithm had to be found which transforms a linear representation with a random starting point reliably into a linear representation with a defined starting point. To accomplish this, the lowest index in the lists is taken as a potential starting point for linearisation. If several instances of this lowest index are present in the list, each instance is credited a score according to the subsequent indices in the list. The instance that is followed by the highest count of lowest indices gets the highest score, and the list is rearranged such that this instance becomes the first entry. Lists transformed in this way can then simply be compared using Python's equality operator, so that redundant solutions can be identified and eliminated.

Cre-ACEMBLER has proven to be a valuable, robust tool in extensive testing by users of the Eukaryotic Expression Facility (EEF) at EMBL Grenoble, proving the reliability of the algorithms described above.

Bibliography:

1. Bieniossek, C., Nie, Y., Frey, D., Olieric, N., Schaffitzel, C., Collinson, I., Romier, C., Richmond, T.J., Steinmetz, M.O. & Berger, I. "Automated unrestricted multigene recombineering for multiprotein complex production" *Nature Methods* 6, 447-450. (2009).
2. Fitzgerald, D.J., Berger, P., Schaffitzel, C., Yamada, K., Richmond, T.J. & Berger, I. "Protein complex expression by using multigene baculoviral vectors" *Nature Methods* 3, 1021-1032 (2006)
3. Bieniossek, C., Imasaki, T., Takagi, Y. & Berger, I. "MultiBac: Expanding the research toolbox for multiprotein complexes", *Trends Biochem. Sci*. 37, 49-57 (2012)
4. Weisstein, E.W. Circular Permutation - From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/CircularPermutation.html
5. Becke, C. "New expression tools for structural analysis of protein-RNA complexes" Masters' Thesis, EMBL / Freie Universität Berlin (2010)
6. Python programming language. http://python.org/
7. The GIMP Toolkit. http://www.gtk.org/
8. PyGTK: GTK+ for Python. http://www.pygtk.org/
9. Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A. *et al*. "Biopython: freely available Python tools for computational molecular biology and bioinformatics." *Bioinformatics* 25, 1422-3(2009)
10. Davis, M.W. Ape - A Plasmid Editor. http://www.biology.utah.edu/jorgensen/wayned/ape/
11. Invitrogen Vector NTI. http://www.invitrogen.com/