Supplemental Material 1

MAKERGAUL: An innovative MAK2- based model and software for real-time PCR quantification

Christoph André Bultmann and Ralf Weiskirchen

Assumptions of the MAKERGAUL-Model

The MAKERGAUL-Model serves to determine DNA concentrations in real-time PCR experimentation. It is a mechanistically calculation model which is based on the following six assumptions:

1. Theoretical Assumptions

***Assumption 1**:* The educts of the DNA polymerase is a DNA single strand, the product a double strand. In each cycle of the PCR each DNA strand can be maximal doubled.

$$(1) \quad DNA_{cycle\ n} = DNA_{cycle\ n-1} + DNA_{cycle\ n-1}$$

***Assumption 2**:* The generation of novel strands is dependent on the content of free DNA polymerase.

$$(2) \quad DNA_{cycle\ n} = DNA_{cycle\ n-1} + DNA_{cycle\ n-1}\ x\ \frac{Enzyme_{usable\ n}}{Enzyme_{blocked\ n}}$$

***Assumption 3**:* The higher the DNA concentration, the more DNA polymerase is occupied by the DNA during the cycle and the less often a DNA strand encounters a free polymerase that copies it.

$$\rightarrow(3) \quad DNA_{cycle\ n} = DNA_{cycle\ n-1} + DNA_{cycle\ n-1}\ x\ \frac{Enzyme_{usable\ n}}{Enzyme_{usable\ n} + DNA_{cycle\ n-1}}$$

***Assumption 4**:* DNA polymerases bind with a certain probability also to double stranded DNA [1]. As a consequence, they are no longer available in the replication phase and the amount of usable polymerases decreases after each cycle.

$$(4) \quad Enzyme_{usable\ n} = \frac{Enzyme_{usable\ n-1}}{1 + DNA_{cycle\ n-1}\ x\ Factor_{inhibition}}$$

***Assumption 5**:* The measured fluorescence in real-time PCR corresponds to the DNA concentration, along with the basis fluorescence in the measurement system [2].

$$(5) \quad Fluorescence_{cycle} = DNA_{cycle} + Fluorescence_{baseline}$$

***Assumption 6**:* Primer deficiency plays no role in slowing down the reaction in the late phase of the PCR.

1.1 Theoretical similarities and differences between MAKERGAUL and MAK2

Assumptions 1 (the principle of DNA-duplicating in a perfect PCR-cycle) and 5 (a fluorescence signal contains baseline and DNA-fluorescence) are also part of the MAK2-model [1]. The main changes in the MAKERGAUL algorithm in comparison to MAK2 are the introduction of a concentration for usable DNA-polymerase and its inhibition by the DNA-concentration (Assumptions 2 to 4). Assumption 6 is also part of MAK2 in which it limits the observed cycles for quantification to the exponential phase of the PCR [1]. We follow P. Kainz who demonstrated that the blockade of the DNA-polymerase is the main factor for the PCR plateau phase [2] and ignore primer concentrations for choosing the cycles included in our analysis.

2. Software implementation

2.1 Fitness-value

Formulas (3), (4) and (5) are the basis for a recursive model that incorporates the parameters DNA, $Enzyme_{usable}$, $Factor_{inhibition}$ und $Fluorescence_{baseline}$ at the beginning of the PCR (*cycle 0*) and calculates them in the following steps.
Parameter optimization is performed by a Downhill-Simplex algorithm [2].
The necessary evaluation of the fitness parameter for this is a function that computes the sum of squared relative errors (SSRE) for the differences between measured values and calculated model curve.
This function was chosen as a superior alternative to the more common sum of absolute squared errors (SSE). Using SSE for curve fitting, differences in later cycles are higher weighted than in the more important, early cycles of PCR (high absolute fluorescence values = high absolute differences; low absolute fluorescence values = low absolute differences).
It is also possible to compare SSRE between samples with different levels of maximal fluorescence as a precision-marker of the analysis.

2.2 Limits

For all 4 parameters of the model, we define limit values: (i) $DNA_{cycle\ 0}$ was chosen between 1E-14 and 0.1. (ii) For $Enzyme_{usable\ 0}$ and $Factor_{inhibition}$ there exists no theoretical assumption about their ranges. Therefore, the limits were set empirically to 0.0001 and double of the maximal fluorescence ($Enzyme_{usable}$ = 0.0009 and $Factor_{inhibition}$ = 1), respectively. For baseline-fluorescence we chose the 1.9-fold fluorescence of the first measured value in both directions.

2.3 Limit Control

The constraints of the function are controlled in different ways. Due to the lack of computation-speed in the PHP-implementation, only a Random-Box-Restriction [4] was applied. Seeding of all points takes part in the whole parameter interval. In the C++-implementation, the seeding* of the points is restricted to the area of the actually best result (at the beginning of an iteration loop) or to the area of the actual best point (for correcting points inside an iteration step), respectively. This leads to an overall faster performance but unfortunately also to weaker borders. Especially the violation of the lower limit of $Factor_{inhibition}$ creates a local minimum at which the simplex often

gets stuck. To solve this problem, we have incorporated a fitness-penalty [4] and $Factor_{inhibition}$ < minimum was implemented in the MAKERGAUL-function.
(* The term seeding defines the random setting of starting values for the variables of a function.)

2.4 Excludes

For choosing measurement-data that should be included in computation, we define some general rules:

2.4.1 No including of declining fluorescence values

At the end of real-time-PCR, fluorescence-levels often begin to decline. These data is excluded.

2.4.2 No measurement possible

If the maxima of fluorescence in all cycles do not exceed the twofold maximal baseline-fluorescence, no duplication can be detected. The result is then “not computable”, most likely because the $DNA_{cycle\ 0}$ is zero.

2.5 The MAKERGAUL-Algorithm

In the following, a pseudo C++-code of the implementation used in MAKERGAUL and makergaulxx is presented. Please note that the fitness-penalty for $Factor_{inhibition}$ is not shown because it is not part of the modelling-computation.

Code 1: Pseudo C++-code of the MAKERGAUL-fitness-implementation

```
/*Computes the fitness of a given Downhill-Simplex-Point by using the MAKERGAUL-algorithm
and SSRE between measured and computed values*/
MAKERGAUL_fitness(dna[0],enzyme_usable[0],factor_inhibition,fluorescence_baseline)
  {
    //First computed DNA-value for later curve-fitting is DNA n=0 + baseline-fluorescence
    computed_value[0] = dna[0] + fluorescence_baseline;
    //First value in the sum of computed fluorescence-data is the first computed data
    computed_value_sum = computed_value[0];

    //MAKERGAUL-model: Generating data
    for(n = 1; n <= cycles; n++)
      {
        //First step: Compute the available polymerase for this cycle
        e_usable[n] = e_usable[n-1]/(1 + e_usable[n-1]*dna[n-1]);
        //Second step: Compute the dna for this cycle
        dna[n] = dna[n-1] + dna[n-1] * e_usable[n]/(e_usable[n]+dna[n-1]);
        //Third step: Compute te model-fluorescence for this
        computed_value[n] = dna[n] + fluorescence_baseline;
        //Fourth step: Add the actual fluorescence to the sum of computed fluorescence
        computed_value_sum += computed_value[n];
      }

    //Averaging model-values
    computed_value_mean = computed_value_sum / cycles;

    //SSRE: Curve fitting
    for(n = 0; n <= cycles; n++)
      {
        //First step: relative error for the actual cycle
        ssre_step = (measured_value[n] - computed_value[n]) /
                (measured_value[n] - computed_value_mean);
        //Second step: squaring
        ssre_step = ssre_step^2;
        //Third step: Add SRE to SSRE
        ssre += ssre_step,
      }
    //Return SSRE to Downhill-Simplex-function
```

```
    return(ssre);
  }
//PSEUDO C++-CODE! DOES NOT WORK IN A REAL PROGRAM!
//(No data-types, wrong parameter in header, wrong array access,...)
```

References cited:

[1] P. Kainz, The PCR plateau phase - towards an understanding of its limitations, Biochim. Biophys. Acta 1494 (2000) 23-27.

[2] A. Nelder, R. Mead, A simplex method for function minimization, Comput. J. 7 (1965) 308-313.

[3] G.J. Boggy, P.J. Woolf, A mechanistic model of PCR for accurate quantification of quantitative PCR data, PLoS ONE 5(8) (2010) e12355.

[4] F. Le Floc'h, Issues of Nelder-Mead simplex optimisation with constraints (January 2, 2012), available at SSRN: http://dx.doi.org/10.2139/ssrn.2097904.