数理解析研究所講究録 1465 巻 2006 年 138-150

A positive detecting algorithm for DNA library screening based on CCCP

慶應大学・理工学研究科　上原 啓明 (Hiroaki Uehara)
Graduate School of Science and Technology,
Keio University

Abstract

We describe a new algorithm for extracting as much information as possible from pooling experiments for library screening based on concave-convex procedure (CCCP). In this paper, we introduce a new positive clone detecting algorithm, CCCP pool result decoder (CCPD). The performance of our CCPD is compared, by simulation, with Bayesian network pool result decoder (BNPD) proposed by Uehara *et al.* and Markov chain pool result decoder (MCPD) proposed by Knill *et al.* (1996).

Key words: DNA library screening, pooling experiment, two stage test, group testing, Bayesian network, CCCP.

1 Introduction

In a screening experiment, it is desired to determine whether a clone contains a specific sequence of nucleotides. The clone is positive if it contains the specific sequence. A goal of DNA library screening is to identify all positive clones. Group testing is utilized to reduce the number of tests. In group testing, the plural clones are assayed in groups. Each group is called a *pool*. If a pool gives a negative outcome, all clones contained in it are found to be negative. In this case, we can save the number of tests. On the other hand, if a pool is positive, we know that the pool contains at least one positive clone. This screening method is called a pooling experiment which is a popular group testing.

The aim of a pooling experiment is to save the number of tests to find positive clones among a large amount of clones. However, the existence of errors cannot be disregarded. That is, we should take into account of the possibility of false positive and false negative. Therefore, in pooling experiments it is required that positive clones can be detected with a high probability even when errors exist in experiments. In this paper, we introduce a new algorithm based on concave-convex procedure (CCCP) and show the efficiency of the algorithm.

2 Pooling experiments and their stochastic model

In this section, we describe the outline of a pooling design and its stochastic model treated in this paper.

2.1 Pooling designs

Here, we explain a pooling design which is a method for planning a group testing. In pooling experiments, each pool includes many kinds of clones. A collection of pools which includes various combinations of clones is called a *pooling design*. When there are quite few positive clones, it may be possible to reduce the number of tests by designing an efficient pooling experiment.

We denote clones by c and pools by G. Let C be the set of n clones and let $\mathcal{G}$ be a collection of m pools. Each pool G in $\mathcal{G}$ is a subset of C corresponding to clones in the pool.

The incidence relation of C and $\mathcal{G}$ is written by an $m \times n$ $\{0,1\}$-matrix or a bipartite graph. This combinatorial structure is said to be a *pooling design*.

Let $E = \{\{c, G\} \mid c \in G\}$. Then, we call the bipartite graph $(C, \mathcal{G}; E)$ a *Tanner graph*. The name of "Tanner graph" comes from the field of coding theory. Here, we utilize the same name to make our algorithm correspond to that of low density parity check (LDPC) code.

2.2 A stochastic model of pooling experiments

Let X_c be a random variable such that

$$X_c = \begin{cases} 0, & \text{if clone } c \text{ is negative,} \\ 1, & \text{if clone } c \text{ is positive.} \end{cases}$$

Usually, the priori probability $P(X_c = 1)$ is small, for example, $P(X_c = 1) = 0.0001$, 0.001, etc. And, let Z_G be a random variable defined by

$$Z_G = \bigvee_{c \in G} X_c, \tag{1}$$

where $\bigvee$ implies the or-sum of X_c's in G. If a pool G includes only negative clones, then $Z_G = 0$. And, if a pool G includes at least one positive clone, then $Z_G = 1$. Let S_G be a random variable representing the observation of experiment for pool G. An observation of response for a screening test is often represented by the following four level values (see

[1]):

$$S_G = \begin{cases} 0, & \text{if pool } G \text{ is negative,} \\ 1, & \text{if pool } G \text{ is weak positive,} \\ 2, & \text{if pool } G \text{ is medium positive,} \\ 3, & \text{if pool } G \text{ is strong positive.} \end{cases}$$

This is because the response of the experiment is measured by a fluorescence sign by a machine. Then we should take into account of the error probability $P(S_G = s \mid Z_G = z)$, i.e. $P(S_G = 1, 2, 3 \mid Z_G = 0)$ is the probability of false positive, while $P(S_G = 0 \mid Z_G = 1)$ is that of false negative. Here, we assume that X_c's are independent, each observation S_G depends only on Z_G, and observation S_G's are independent under the condition that Z_G's are known.

Let $\boldsymbol{X} = (X_{c_1}, \ldots, X_{c_n})$ and $\boldsymbol{S} = (S_{G_1}, \ldots, S_{G_m})$ be random vectors. In an experiment, when observation $\boldsymbol{S}$ is measured, posterior probability is written by

$$\begin{aligned} P(\boldsymbol{X} = \boldsymbol{x} \mid \boldsymbol{S} = \boldsymbol{s}) &= \frac{P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{S} = \boldsymbol{s})}{P(\boldsymbol{S} = \boldsymbol{s})} \\ &= KP(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{S} = \boldsymbol{s}) \\ &= KP(\boldsymbol{X} = \boldsymbol{x})P(\boldsymbol{S} = \boldsymbol{s} \mid \boldsymbol{X} = \boldsymbol{x}) \\ &= K \prod_{c \in C} P(X_c = x_c) \prod_{G \in \mathcal{G}} P(S_G = s_G \mid Z_G = z_G), \end{aligned}$$

where, $K = P(\boldsymbol{S} = \boldsymbol{s})^{-1}$ is a constant and $z_G = \bigvee_{c \in G} x_c$.

Therefore, the marginal conditional probability for X_c is

$$P(X_c = x \mid \boldsymbol{S} = \boldsymbol{s}) = K \sum_{x_c = x} \prod_{d \in C} P(X_d = x_d) \prod_{G \in \mathcal{G}} P(S_G = s_G \mid Z_G = z_G),$$

where $\sum_{x_c = x}$ means the sum of all $\boldsymbol{x} \in \{0, 1\}^n$ such that $x_c = x$. Moreover, it is assumed that the values $P(S_G = s_G \mid Z_G = z_G)$ are empirically known from the previous experiments.

Note that we need 2^{n-1} summations to calculate the probability $P(X_c = x \mid \boldsymbol{S} = \boldsymbol{s})$. Hence, as the number of clones n becomes large, as the amount of calculation increase exponentially. Therefore, an efficient algorithm which calculates this value is necessary. In 1996, [1] proposed an algorithm based on Markov Chain Monte Carlo method. In 2005, [3] proposed an algorithm based on Bayesian network. In this paper, we will propose another efficient and faster algorithm.

3 Positive clone detecting algorithm

In this section, we describe our new algorithm based on concave-convex procedure (CCCP) [5].

3.1 Bethe free energy

$P(\boldsymbol{X} = \boldsymbol{x} \mid \boldsymbol{S} = \boldsymbol{s})$ can be expressed as a marginal distribution of the following conditional joint distribution $P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{U} = \boldsymbol{u} \mid \boldsymbol{S} = \boldsymbol{s})$:

$$\begin{aligned} &P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{U} = \boldsymbol{u} \mid \boldsymbol{S} = \boldsymbol{s}) \\ &= K \prod_{G \in \mathcal{G}} \Big(P(S_G = s_G \mid Z_G = \bigvee_{c \in G} u_c^G) \prod_{c \in G} \delta(u_c^G, x_c) \Big) \prod_{c \in C} P(X_c = x_c), \end{aligned}$$

where $\boldsymbol{u}_G = (u_c^G)_{c \in G} \in \{0,1\}^{|G|}$, $\boldsymbol{u} = (\boldsymbol{u}_G)_{G \in \mathcal{G}} \in \prod_{G \in \mathcal{G}} \{0,1\}^{|G|}$ and

$$\delta(a,b) = \begin{cases} 1, & \text{if } a = b, \\ 0, & \text{if } a \neq b. \end{cases}$$

Then we have

$$P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{U} = \boldsymbol{u} \mid \boldsymbol{S} = \boldsymbol{s}) = \begin{cases} P(\boldsymbol{X} = \boldsymbol{x} \mid \boldsymbol{S} = \boldsymbol{s}), & \text{if } \boldsymbol{u}_G = \boldsymbol{x}_G \text{ for } G \in \mathcal{G}, \\ 0, & \text{otherwise,} \end{cases}$$

where $\boldsymbol{x}_G = (x_c)_{c \in C} \in \{0,1\}^{|G|}$.

Now we define

$$\begin{aligned} &\psi_c(x_c) = P(X_c = x_c) && \text{for } x_c \in \{0,1\} \quad (c \in C), \\ &\psi_G(\boldsymbol{u}_G) = P(S_G = s_G \mid Z_G = \bigvee_{c \in G} u_c^G) && \text{for } \boldsymbol{u}_G \in \{0,1\}^{|G|} \quad (G \in \mathcal{G}), \\ &\psi_{cG}(x_c, \boldsymbol{u}_G) = \delta(u_c^G, x_c) && \text{for } (x_c, \boldsymbol{u}_G) \in \{0,1\} \times \{0,1\}^{|G|} \quad (c \in G). \end{aligned}$$

Then

$$P(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{U} = \boldsymbol{u} \mid \boldsymbol{S} = \boldsymbol{s}) = K \prod_{c \in G} \psi_{cG}(x_c, \boldsymbol{u}_G) \prod_{c \in C} \psi_c(x_c) \prod_{G \in \mathcal{G}} (x_c).$$

For $P(\boldsymbol{X}=\boldsymbol{x},\boldsymbol{U}=\boldsymbol{u}\mid\boldsymbol{S}=\boldsymbol{s})$, the Bethe free energy becomes

$$\begin{aligned}F_B(\boldsymbol{q}) =& \sum_{c\in G}\sum_{(x_c,\boldsymbol{u}_G)\in\{0,1\}\times\{0,1\}^{|G|}} q_{cG}(x_c,\boldsymbol{u}_G)\log\frac{q_{cG}(x_c,\boldsymbol{u}_G)}{\phi_{cG}(x_c,\boldsymbol{u}_G)}\\ &-\sum_{c\in C}(|(c)|-1)\sum_{x_c\in\{0,1\}} q_c(x_c)\log\frac{q_c(x_c)}{\psi_c(x_c)}\\ &-\sum_{G\in\mathcal{G}}(|G|-1)\sum_{\boldsymbol{u}_G\in\{0,1\}^{|G|}} q_G(\boldsymbol{u}_G)\log\frac{q_G(\boldsymbol{u}_G)}{\psi_G(\boldsymbol{u}_G)}\end{aligned}$$

where (c) denotes the set of pools which includes a clone c,

$$\phi_{cG}(x_c,\boldsymbol{u}_G)=\psi_c(x_c)\psi_G(\boldsymbol{u}_G)\psi_{cG}(x_c,\boldsymbol{u}_G)$$

and $q_c(x_c)$, $q_G(\boldsymbol{u}_G)$ and $q_{cG}(x_c,\boldsymbol{u}_G)$ satisfy

$$\begin{aligned}&\sum_{x_c\in\{0,1\}} q_c(x_c)=1 && (c\in C),\\ &\sum_{\boldsymbol{u}_G\in\{0,1\}^{|G|}} q_G(\boldsymbol{u}_G)=1 && (G\in\mathcal{G}),\\ &\sum_{x_c\in\{0,1\}} q_{cG}(x_c,\boldsymbol{u}_G)=q_G(\boldsymbol{u}_G) && \text{for } \boldsymbol{u}_G\in\{0,1\}^{|G|}\quad (c\in G),\\ &\sum_{\boldsymbol{u}_G\in\{0,1\}^{|G|}} q_{cG}(x_c,\boldsymbol{u}_G)=q_c(x_c) && \text{for } x_c\in\{0,1\}\quad (c\in G).\end{aligned}$$

In the remaining part of this paper, we assume that $\psi_c(x_c)$ is always non-zero to simplify the discussions and descriptions. Then the Bethe free energy is simplified into

$$\begin{aligned}F_B(\boldsymbol{q}) =& \sum_{c\in C}\sum_{x_c\in\{0,1\}} q_c(x_c)\log\frac{q_c(x_c)}{\psi_c(x_c)}+\sum_{G\in\mathcal{G}}\sum_{\boldsymbol{u}_G\in\{0,1\}^{|G|}} q_G(\boldsymbol{u}_G)\log\frac{q_G(\boldsymbol{u}_G)}{\psi_G(\boldsymbol{u}_G)}\\ &-\sum_{c\in C}|(c)|\sum_{x_c\in\{0,1\}} q_c(x_c)\log q_c(x_c)\end{aligned}\tag{2}$$

and

$$\begin{aligned}&\sum_{x_c\in\{0,1\}} q_c(x_c)=1 && (c\in C),\\ &\sum_{\boldsymbol{u}_G\in\{0,1\}^{|G|}} q_G(\boldsymbol{u}_G)=1 && (G\in\mathcal{G}),\\ &\sum_{\substack{\boldsymbol{u}_G\in\{0,1\}^{|G|}\\ \text{s.t. } u_c^G=x_c}} q_G(\boldsymbol{u}_G)=q_c(x_c) && \text{for } x_c\in\{0,1\}\quad (c\in G).\end{aligned}\tag{3}$$

3.2 Positive clone detecting algorithm

CCCP is a procedure that finding $\boldsymbol{q}$ minimized Bethe free energy (2) subject to the linear constrains (3) by utilizing Lagrange multiplier method.

Our new positive clone detecting algorithm consists of the following steps.

Step 0 For each clone $c \in C$, set $P(X_c = x)$. For example, $P(X_c = 1) = 0.001,\ 0.002$ and $P(X_c = 0) = 0.999,\ 0.998$.

Step 1 (Initialization of Q): For each $c \in C$, initialize

$$Q_x(c) = P(X_c = x), \qquad x = 0, 1.$$

Step 2.1 (Initialization of Γ and Λ): For each $c \in C$ and $G \in \mathcal{G}$, initialize

$$\Gamma(c) = 1, \qquad \Gamma(G) = 1.$$

For each $(c, G) \in C \times \mathcal{G}$ such that $c \in G$, let

$$\Lambda_x(c, G) = 1, \qquad x = 0, 1.$$

Step 2.2 (Update of Λ): For each $(c, G) \in C \times \mathcal{G}$ such that $c \in G$, update $\Lambda_x(c, G)$ by

$$\begin{cases} R_1(c, G) = P(s_G \mid 1) \prod\limits_{c' \in G \setminus \{c\}} \sum\limits_{x \in \{0,1\}} \Lambda_x(c', G), \\ R_0(c, G) = \big(P(s_G \mid 0) - P(s_G \mid 1)\big) \prod\limits_{c' \in G \setminus \{c\}} \Lambda_0(c', G) + R_1(c, G), \end{cases}$$

$$\Lambda_x(c, G) = \sqrt{\frac{\Gamma(G)}{\Gamma(c)} \frac{e^{|(c)|} P(X_c = x) Q_x(c)^{|(c)|}}{\prod_{G' \in (c) \setminus \{G\}} \Lambda_x(c, G')} \frac{1}{R_x(c, G)}}, \qquad x = 0, 1.$$

Step 2.3 (Update of Γ): For each $c \in C$, update $\Gamma(c)$ by

$$\Gamma(c) = \frac{1}{e} \sum_{x \in \{0,1\}} \frac{e^{|(c)|} P(X_c = x) Q_x(c)^{|(c)|}}{\prod_{G \in (c)} \Lambda_x(c, G)}.$$

For each $G \in \mathcal{G}$, update $\Gamma(G)$ by

$$\Gamma(G) = \frac{1}{e}\Big(\big(P(s_G \mid 0) - P(s_G \mid 1)\big) \prod_{c \in G} \Lambda_0(c, G) + P(s_G \mid 1) \prod_{c \in G} \sum_{x \in \{0,1\}} \Lambda_x(c, G)\Big).$$

Step 2.4 (Iteration of inner loop): Iterate Step 2.2 and Step 2.3 for several times.

Step 3 (Update of Q): For each $c \in C$

$$Q_x(c) = \frac{1}{e}\frac{1}{\Gamma(c)}\frac{e^{|(c)|}P(X_c = x)Q_x(c)^{|(c)|}}{\prod_{G\in(c)}\Lambda_x(c,G)}.$$

Step 4 (Iteration of outer loop): Iterate Step 2.1 and Step 3 for several times until the values Q_x converges.

Step 5 (Output of marginal probability): Output $Q_x(c)$ as the result.

We call our algorithm *concave-convex procedure pool result decoder* (CCPD). If a Tanner graph has no cycle, it is known that each estimate of CCPD and Bayesian network pool result decoder (BNPD) converges to the correct marginal probability after enough iteration. If it has cycles, CCCP can obtain the better estimate than BNPD.

4 Simulation results

4.1 The method of simulations

Our simulation is pursued as follows:

(i) We set the priori probability that each clone is positive and the number of positive clones. We choose a given number of positive clones randomly.

(ii) The experimental results S_G's are determined randomly according as the true value Z_G's and the conditional probability $P(S_G = a \mid Z_G = b)$. These conditional probabilities are given beforehand.

(iii) The experimental results obtained in (ii) are passed to each of BNPD and MCPD algorithms, and the probability of positiveness is computed for each clone. And clones are sorted by the value of $P(X_c = 1 \mid \boldsymbol{S} = \boldsymbol{s})$.

4.2 Simulation 1: a comparison of CCCP, BNPD and MCPD

Firstly, we compare the detectability of true positive clones among three algorithms CCCP, BNPD and MCPD.

Let $n = 1298$ and $m = 131$. The number $n = 1298$ is chosen to compare the experiment with that of [1], which does not satisfy the "unique collinearity condition". In this section we utilize a pooling design satisfying the unique collinearity condition so that it is expected that BNPD algorithm converges. The case when the unique collinearity condition is not satisfied is treated in the later section.

In order to utilize a pooling design satisfying the unique collinearity condition we introduce a combinatorial notion of "packing design". Let $v \geq k$. A (v,k)-*packing* is a

pair $(V, \mathcal{B})$, where V is a v-set of elements (called *points*) and $\mathcal{B}$ is a collection of k-subsets of V (called *blocks*), such that every pair of V occurs in at most one block in $\mathcal{B}$. In our simulation a clone corresponds to a block, and a pool corresponds to a point. A packing design satisfies the unique collinearity condition by its definition, i.e. any two clones are included together at most one pool and a Tanner graph of the design has no loops of length four. Here, we utilize a $(131, 4)$-packing, which can be generated by the finite field $GF(131)$. This pooling design has $n = 1298$ clones and $m(= v) = 131$ pools.

In our simulation, the priori probability each clone being positive is set to $P(X_c = 1) = 0.002$. And we considered four cases when the number of positive clones are from one to four. For given number of true positive clones, we choose true positive clones randomly from 1298 clones. The conditional probability of false-positive or false-negative for the experiment is given from the result of an actual DNA library screening by [1]. They fixed the conditional probability as follows:

$$\begin{aligned} &P(S_G = 0 \mid Z_G = 0) = 0.871, \quad P(S_G = 0 \mid Z_G = 1) = 0.05, \\ &P(S_G = 1 \mid Z_G = 0) = 0.016, \quad P(S_G = 1 \mid Z_G = 1) = 0.11, \\ &P(S_G = 2 \mid Z_G = 0) = 0.035, \quad P(S_G = 2 \mid Z_G = 1) = 0.27, \\ &P(S_G = 3 \mid Z_G = 0) = 0.078, \quad P(S_G = 3 \mid Z_G = 1) = 0.57. \end{aligned} \tag{4}$$

The reader may consider that the error probabilities in (4) do not seem to be natural because the monotoniety

$$P(S_G = 0 \mid Z_G = 0) > P(S_G = 1 \mid Z_G = 0) > P(S_G = 2 \mid Z_G = 0) > P(S_G = 3 \mid Z_G = 0)$$

is not satisfied. Thus we also treated the following artificially settled value of errors:

$$\begin{aligned} &P(S_G = 0 \mid Z_G = 0) = 0.856, \quad P(S_G = 0 \mid Z_G = 1) = 0.021, \\ &P(S_G = 1 \mid Z_G = 0) = 0.126, \quad P(S_G = 1 \mid Z_G = 1) = 0.154, \\ &P(S_G = 2 \mid Z_G = 0) = 0.016, \quad P(S_G = 2 \mid Z_G = 1) = 0.288, \\ &P(S_G = 3 \mid Z_G = 0) = 0.002, \quad P(S_G = 3 \mid Z_G = 1) = 0.537. \end{aligned} \tag{5}$$

This probability is given by

$$\begin{aligned} &P(S_G = 1 \mid Z_G = 0) = q, && P(S_G = 1 \mid Z_G = 1) = q'^3, \\ &P(S_G = 2 \mid Z_G = 0) = q^2, && P(S_G = 2 \mid Z_G = 1) = q'^2, \\ &P(S_G = 3 \mid Z_G = 0) = q^3, && P(S_G = 3 \mid Z_G = 1) = q' \end{aligned}$$

and

$$P(S_G = 0 \mid Z_G = 0) = 1 - q - q^2 - q^3,$$
$$P(S_G = 0 \mid Z_G = 1) = 1 - q' - q'^2 - q'^3,$$

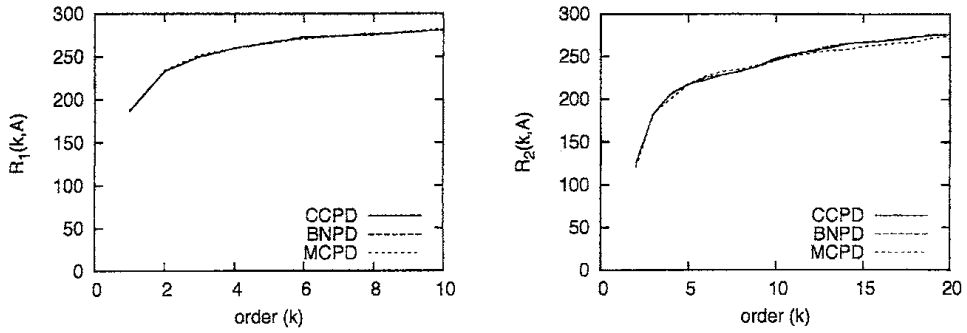
where $q = 0.126$ and $q' = 0.537$. The value q and q' is set so that the probabilities are similar to those of (4).

Firstly, we adopt error probability (4). Then under these conditions, experimental result for each pool is randomly decided according to the conditional probability (4) and Z_G's. Note that given number of positive clones are randomly chosen beforehand and the values Z_G's are determined by equation (1).

Let $R_d(k, A)$ be the number of simulations such that all d positive clones are ranked within the top k-th position when algorithm A is utilized.

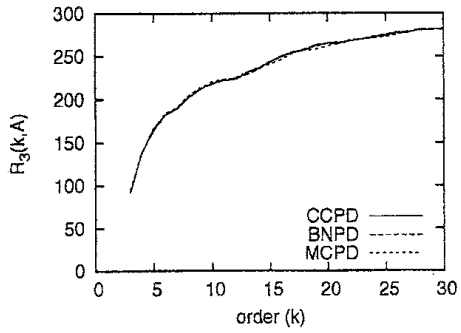
In the real pooling experiment, after knowing the result of each pool, we proceed to the second stage, that is, clones are tested from the highest rank to the lower rank. Thus in order to reduce the number of individual experiment in the second stage, it is desired to obtain a good $R_d(k, A)$. That is, $R_d(k, A)$ may differ not only by the pooling experiment but also by the decoding algorithm A. If $R_d(k, A_1) \geq R_d(k, A_2)$ holds for any k, then algorithm A_1 is considered to be better than A_2. We will compare $R_d(k, \mathrm{CCPD})$ with $R_d(k, \mathrm{BNPD})$ and $R_d(k, \mathrm{MCPD})$.

Figure 1 includes graphs of $R_d(k, \mathrm{CCPD})$, $R_d(k, \mathrm{BNPD})$ and $R_d(k, \mathrm{MCPD})$. We can see that detectability of positive clones of CCPD, BNPD and MCPD is almost same in this simulation.

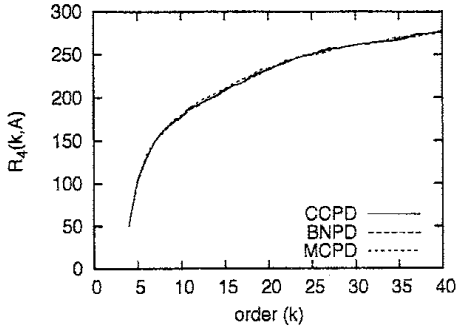


(1) number of positive clones: 1

(2) number of positive clones: 2



(3) number of positive clones: 3



(4) number of positive clones: 4

Figure 1: Detectability of positive clones ($n = 1298$, $m = 131$)

4.3 Simulation 2: a comparison of execution speed

Here, we should note that the execution speed of CCPD is about ten times faster than that of MCPD and about thirty times slower than that of BNPD as is shown in Table 1 and Figure 2. Note that, the vertical axis of Figure 2 is logarithmic scale. Moreover, execution speed for both algorithms are proportional to the number of clones.

Table 1: Execution time

unit : second

n	CCPD	BNPD	MCPD
981	0.22	0.00	1.91
1298	0.27	0.01	2.49
3088	0.81	0.02	8.49
6371	1.68	0.05	17.60
10121	3.33	0.09	27.73
30050	11.09	0.26	81.80

CPU: Intel® Xeon™ 3.06 GHz

OS: Red Hat Linux 9

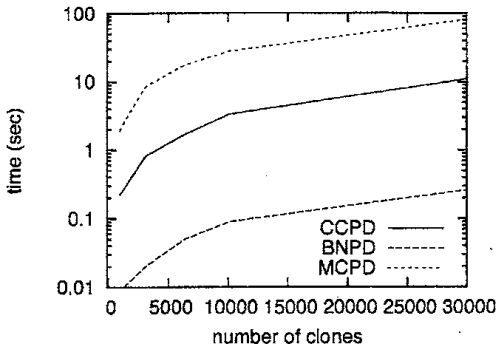
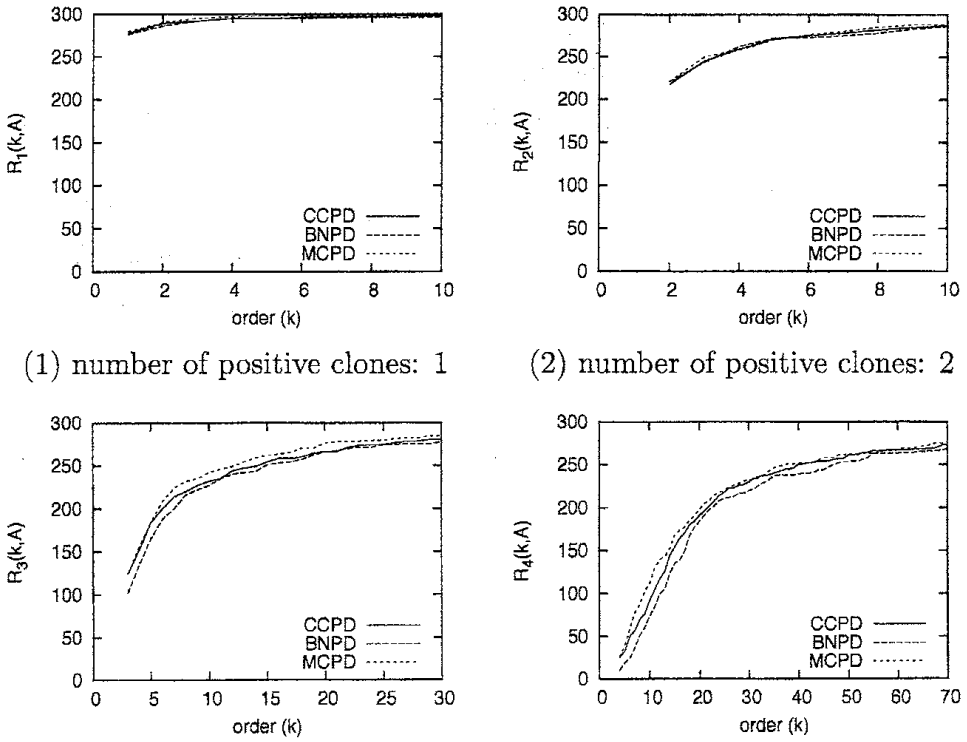


Figure 2: Execution time

4.4 Simulation 3: non unique collinearity condition case

In this section we consider the case when the unique collinearity condition is not satisfied. We pursued the simulation on the following condition. The number of clones $n = 1298$ in the first simulation is chosen so that we can compare the result with the real pooling experiment which is reported in [1] for 1298 kinds of clones of Human chromosome 16 with 47 pools. In the experiment they utilized a random four-set pooling design. A random four-set design puts each clone in four pools. The design does not satisfy the "unique collinearity condition", i.e. there are two clones which are included together in more than two pools, hence a Tanner graph of the design has loops of length four. In order that a pooling design with 1298 clones and replication number four satisfies the unique collinearity condition, we need at least 131 pools since it must be a packing design. In this section we utilize the same pooling design with 47 pools and try 300 simulations to exam the ability of CCPD algorithm. And we adopt error probability (5) instead of (4).

In this case, we may not expect a good performance for BNPD algorithm since the pooling design does not satisfy the unique collinearity condition. However, CCPD algorithm has better performance than BNPD algorithm from Figure 3.



(1) number of positive clones: 1 (2) number of positive clones: 2

(3) number of positive clones: 3 (4) number of positive clones: 4

Figure 3: Detectability of positive clones in the case of error probability (5) ($n = 1298$, $m = 47$)

5 Conclusion

As a conclusion, for a pooling design a packing design should be utilized to assure the unique collinearity condition. In the case when we can utilize a packing design, firstly BNPD algorithm should be applied, then we will get good detectability of positive clones with a high execution speed. If we cannot utilize a packing design, CCCP algorithm is relatively efficient at short time.

References

[1] E. Knill, A. Schliep and D. C. Torney (1996), Interpretation of pooling experiments using the Markov chain Monte Carlo method, *Journal of Computational Biology*, **3**, 395-406.

[2] T. Shibuya, K. Harada, R. Tohyama and K. Sakaniwa (2005), Iterative Decoding Based on the Concave-Convex Procedure, *IEICE Trans. Fundamentals*, **E88-A(5)**, 1346-1364

[3] H. Uehara and M. Jimbo (2005), A positive detecting code and its decoding algorithm for DNA library screening, *Journal of Computational Biology*, submitted.

[4] J. S. Yedidia, W. T. Freeman and Y. Weiss (2001), Bethe free energy, Kikuchi approximations, and belief propagation algorithms, *Tech. Rep. of Mitsubishi Electric Research Lab.*, **TR-2001-16**.

[5] A. L. Yuille (2002), CCCP algorithms to minimize the Bethe and Kikuchi free energies: convergent alternatives to belief propagation, *Neural Computation*, **14**, 1691-1722.