NIWA
Taihoro Nukurangi

Preparing exact Most Probable Number (MPN) tables using occupancy theory, and accompanying measures of uncertainty

Graham B. McBride

NIWA Technical Report 121
ISSN 1174-2631
2003

Preparing exact Most Probable Number (MPN) tables using occupancy theory, and accompanying measures of uncertainty

Graham B. McBride

NIWA Technical Report 121
2003

Published by NIWA
Wellington
2003

Edited and produced by
Science Communication, NIWA,
Private Bag 14901, Wellington, New Zealand

ISSN 1174-2631
ISBN 0-478-23266-7

© NIWA 2003

Citation:
McBride, G.B. (2003).
Preparing exact Most Probable Number (MPN) tables using occupancy theory,
and accompanying measures of uncertainty.
NIWA Technical Report 121. 62 p.

Disclaimer

Notwithstanding any mention, recommendation, or lack of recommendation for a product or service made in this report by NIWA, NIWA will not be held to have made any endorsement, warranty, promise, or representation as to the suitability, competence, or performance of that product or service.

The National Institute of Water and Atmospheric Research
is New Zealand's leading provider
of atmospheric, marine,
and freshwater science

Visit NIWA's website at http://www.niwa.co.nz

Contents

Abstract

McBride, G.B. (2003). Preparing exact Most Probable Number (MPN) tables using occupancy theory, and accompanying measures of uncertainty. ***NIWA Technical Report 121.*** **62 p.**

Procedures for calculation of exact 'Most Probable Numbers' (MPNs) using occupancy theory are generalised for laboratory microbiological tests using up to four dilutions. Comparisons are made of exact MPNs with values in tables in compilations of standard methods, which are based on approximate methods and are variously rounded. New, accurate tables can easily be developed, for which examples are given. The relative merits of confidence intervals and credible intervals as measures of uncertainty about these MPN values are discussed. Computer code is developed using accurate calculation algorithms and is listed in the Appendices.

Introduction

In performing microbiological laboratory fermentation tests one uses sets of replicated tubes or wells containing known reducing amounts of innoculum. Typically up to four sets are used, and called a 'dilution series'. For example, one may use three sets of tubes each with five replicates: 5x100 mL, 5x10 mL, and 5x1 mL. The pattern of positive results obtained in such tests is related to a 'most probable number' (MPN) of microorganisms via a standard table (e.g., as contained in APHA 1995), and this is the final result of the test — perhaps accompanied by a confidence interval (also available in those tables). These tables have been worked out over many years, starting with McCrady (1915), and continuing to the present (e.g., Eisenhart & Wilson 1943, Cochran 1950, de Man 1983, Best 1990) using a variety of approximate methods, based on an assumption of random sampling from Poisson distributions (Tillett 1995).

However, if it is desired to use a non-standard dilution series, the laboratory worker faces the problem that a standard table may not be available. How then could it, and some measure of its uncertainty, be constructed? Furthermore, the standard tables that do exist adopt different conventions for rounding the results. For example, consider a case where a 5x100 mL, 5x10 mL, 5x1 mL dilution series has been used and a 5-5-2 pattern of positive tubes was obtained. For this case 'Standard methods for the examination of water and wastewater' 14th edition (APHA 1975) gives the result as MPN = 540 per 100 mL, whereas the 19th edition (APHA 1995) gives it as MPN = 500 per 100 mL. The difference is presumably because a different rounding convention was used, i.e., the number of significant digits retained in the result.[1]

This issue first arose for the author in the evaluation of methods in the 1998 Preliminary Study of the Freshwater Microbiological Programme (FMRP), for MPN enumeration of *Campylobacter* (McBride et al. 2002). In brief, a 1 L sample was delivered to the laboratory, from which 830 mL of subsamples was drawn. These comprised one of 500 mL and three each of 100 mL and of 10 mL. Each of these seven subsamples was filtered and the filtrates placed in test tubes containing a broth. The broth volume was the same in all tubes, regardless of the subsample volume filtered. The tubes were incubated and the number of positive tubes counted for each set. A 1x3x3 MPN table was therefore needed to translate the results of the tubes' incubation to an enumeration. The literature abounds with 3x3x3 and 5x5x5 tables (e.g., McCrady 1918. Swaroop 1938, de Man 1975, 1977, 1983, Beliaeff & Mary 1993). There are also some 1x5x5 tables (e.g., Anon. 1983, WHO 1984, Tillett & Coleman 1985), but no 1x3x3 table had been sighted. Accordingly, I constructed one such table, using Tillett &

---

[1] There are a number of other differences (unexplained) between the tables, some probably the result of different rounding conventions. Just why the 5-3-3 pattern result is given as 180 per 100 mL in APHA (1975) and as 170 per 100 mL in APHA (1995) is unclear. Also the former gives the 5-5-5 result as ≥2400 per 100 mL, whereas the latter gives it as ≥1600 per 100 mL! The latest edition (20th, 1998) contains the same table as does the 19th.

Coleman's method, because it is the most accurate, being based on 'occupancy theory' (David & Barton 1962). It makes no approximating assumptions at all,[2] requiring only that microorganisms be distributed randomly in the samples (this can readily be achieved by thorough mixing). In this report I generalise their result so that any table can be computed, using a computer program ('**XactMPN**') developed herein. Another 1x3x3 MPN table is developed, using different tube volumes (1x670 mL, 3x100 mL, 3x10 mL) such that the entire one litre sample is used.

In the Full Study of the FMRP a 3x3x3 setup was adopted as it gave better precision than the 1x3x3 option. I present a table for this case also, comparing it to approximate tables from the literature. I also develop exact 1x5x5, 5x5x5, and 5x5x5x5 tables, comparing them with other versions — including those found in compilations of standard methods. Tables for the 49x48 two-volume IDEXX Quanti-Tray 2000™ and for the 51 well constant-volume IDEXX Quanti-Tray™ are also generated.

Confidence limits and credible limits for MPN estimates are also discussed. It is noted that because water and health management authorities often need to be concerned with the precision of an MPN result, this is an area deserving of further attention as it appears that further progress could be made to resolve the issues.

Theory

The MPN for a given pattern of results (e.g., 1-2-2 positive tubes out of a total set of 1x3x3) will be derived by repetitive calculation of the probability of getting that pattern for each and every value of bacteria numbers between nominated limits. These are the 'occurrence probabilities'. The distribution of these probabilities over those bacteria numbers is smooth and unimodal, so that the MPN is the number of bacteria at the distribution's mode divided by the sample volume filtered or placed in the tubes.

To develop the required formula we first consider each set of tubes individually.

Considering each set separately

Denote the number of tubes in any fermentation set by R (so R = 1 or 3 in the FMRP Preliminary Study's *Campylobacter* setup). To develop the argument we first consider each set separately and obtain the formula for the probability of getting any number of positive tubes in that set, it being assumed that one bacterium will always cause a tube to return a positive result after incubation.

The formula for this probability comes from 'occupancy theory' and was described by David & Barton 1962[3] (who noted that it flows from Abraham De Moivre's 'Doctrine of chances', first published in 1718). Its development is as follows: if n bacteria are distributed at random among the R replicate test tubes (which contain equal filtrate or sample volumes), the probability that r tubes will receive at least one bacterium (and so show a positive response) is:

$$\Pr(r \mid R,n) = {}^{R}C_{r} \sum_{t=0}^{r} {}^{r}C_{t} (-1)^{t} \left(\frac{r-t}{R} \right)^{n} \quad \text{for } r\ (\le n) = 1, 2, \ldots, R \qquad (1)$$

---

[2] It does not require an assumption of random sampling from a Poisson distribution, as do the approximate methods

[3] Using the equation of David & Barton (1962: 242), altering their nomenclature so that their $N \rightarrow R$.

where ${}^{R}C_r$ (= $R!/\{r![R-r]!\}$) is the number of combinations of R objects taken r at a time, t is a dummy summation index, and the '$|$' symbol denotes 'given'. 'Pr($r \mid R, n$)' means the probability of getting r positive tubes in the set of R tubes, given n bacteria to randomly distribute in that set.

Three special cases (boundary conditions) are not covered by this formula and must be stated separately — they will be needed in the subsequent calculation procedures. First, if there are no bacteria in the set (i.e., $n = 0$) it is certain that no tubes will be positive, i.e.,

$$\Pr(0 \mid R, 0) = 1 \tag{2a}$$

and, conversely, it is impossible for any tube to be positive if there are no bacteria actually present, i.e.,

$$\Pr(r \mid R, 0) = 0 \quad \text{for } r = 1, 2, \ldots, R \tag{2b}$$

Second, it is impossible for there to be no positive tubes if bacteria are actually present, i.e.,

$$\Pr(0 \mid R, n) = 0 \quad \text{for } n \geq 1 \tag{3}$$

Third, it is impossible for the number of positive tubes to exceed the number of bacteria, i.e.,

$$\Pr(r \mid R, n) = 0 \quad \text{for } r\ (\leq R) > n \tag{4}$$

For the general case (i.e., $1 \leq r \leq R$, $n \geq 1$ and $r \leq n$) there are two means of calculating the probabilities given by Equation 1. The first is to use a recursion formula, as did Tillett & Coleman (1985). This arises by noting that the equation can be written as

$$\Pr(r \mid R, n) = \frac{1}{R^n} \frac{R!}{(R-r)!} S_n^{(r)} \tag{5}$$

where $S_n^{(r)}$ (denoted as '$\Delta_{r,n}$' by Tillett & Coleman) is Stirling's number of the second kind, defined as (Roberts 1984)

$$S_n^{(r)} = \frac{1}{r!} \sum_{t=0}^{r} {}^{r}C_t (-1)^t (r-t)^n \tag{6}$$

[The upper summation limit can also be taken as $r-1$, because $(r-t)^n = 0$ when $t = r$.] We can evaluate these numbers using their recurrence relationship (Abramowitz & Stegun 1972):

$$S_n^{(r)} = r S_{n-1}^{(r)} + S_{n-1}^{(r-1)} \tag{7}$$

Now by substituting this relation into Equation 5 and simplifying we obtain the recurrence relation for the probabilities Pr($r \mid R, n$) as

$$\Pr(r \mid R, n) = \frac{r}{R} \Pr(r \mid R, n-1) + \frac{R+1-r}{R} \Pr(r-1 \mid R, n-1):\ r\ (\leq n) = 1, 2, \ldots, R \tag{8}$$

which is the equation given by Tillett & Coleman (1985). This equation can be calculated using the boundary conditions given in Equations 2 and 3. The solution is completed by Equation 4, for all possible cases.

The other approach is to calculate Equation 1 directly, simplifying it (by cancelling out the $r!$ term and grouping constant terms outside the summation symbol) to:

$$\Pr(r|R,n) = \frac{1}{R^n}\frac{R!}{(R-r)!}\sum_{t=0}^{r-1}(-1)^t\frac{(r-t)^n}{t!(r-t)!} \quad \text{for } r\ (\le n) = 1, 2, \ldots, R \tag{9}$$

again using Equations 2–4, as appropriate.

We must now combine the sets of tubes used.

Combining the sets

Consider the series of three fermentation tubes in which we have:

- I tubes each of volume V_I of which i are positive (so $0 \le i \le I$);
- J tubes each of volume V_J of which j are positive (so $0 \le j \le J$);
- K tubes each of volume V_K of which k are positive (so $0 \le k \le K$).

The total volume in the tubes is

$$V = IV_I + JV_J + KV_K \tag{10}$$

and so *for each bacterium* present in the tubes:

- the probability that it is in an I-tube is $p_I = IV_I/V$;
- the probability that it is in a J-tube is $p_J = JV_J/V$;
- the probability that it is in a K-tube is $p_K = KV_K/V$.

These probabilities are, of course, the proportion of total sample volume in each set.

Now let us consider all bacteria in the tubes, and denote their total number by n. We know that the number of ways they may be divided into the three sets of tubes, with n_I in the I-tubes, n_J in the J-tubes, and n_K in the K-tubes is $n!/(n_I!n_J!n_K!)$ (e.g., Freund 1992). This enables us to compute the required occurrence probability — that i, j, and k tubes in the three sets are positive, given that there are n bacteria present. This is computed as the product of the appropriate probabilities summed over all possible numbers of bacteria in each set (i.e., over n_I, n_J, and n_K) such that their total is n, i.e.,

$$\Pr(i,j,k|n) = \sum_{n_I=0}^{n}\sum_{n_J=0}^{n-n_I}\Pr(i|I,n_I)\Pr(j|J,n_J)\Pr(k|K,n_K)\frac{n!}{n_I!n_J!n_K!}p_I^{n_I}p_J^{n_J}p_K^{n_K} \tag{11}$$

where $n_K = n - (n_I + n_J)$. Note that this double sum could become computationally demanding as n gets large.

If there are only two sets of tubes (say I and J), such as in the IDEXX Quanti-Tray/2000™, the occurrence probability is

$$\Pr(i,j|n) = \sum_{n_I=0}^{n}\Pr(i|I,n_I)\Pr(j|J,n_J)\frac{n!}{n_I!n_J!}p_I^{n_I}p_J^{n_J} \quad \text{where } n_J = n - n_I \tag{12}$$

and if there are four sets the occurrence probability is

$$\begin{aligned}\Pr(i,j,k,l|n)=&\sum_{n_I=0}^{n}\sum_{n_J=0}^{n-n_I}\sum_{n_K=0}^{n-(n_I+n_J)}\Pr(i|I,n_I)\Pr(j|J,n_J)\Pr(k|K,n_K)\Pr(l|L,n_L)\\&\times\frac{n!}{n_I!n_J!n_K!n_L!}p_I^{n_I}p_J^{n_J}p_K^{n_K}p_L^{n_L}\end{aligned}\qquad(13)$$

If there is only one set of tubes, such as in the 51-well IDEXX Quanti-Tray™, the occurrence probability is simply given by Equation 8 or 9 (setting $r = i$), i.e., by $\Pr(i\,|\,R, n)$.

Simpler (approximate) equations

Equations 1–13 involve considerable computational complexity. In great contrast is the disarmingly simple procedure given (without proof) by Thomas (1942), i.e.,

$$\text{MPN (per 100 mL)} = 100\frac{P}{\sqrt{NT}}\qquad(14)$$

where P is the number of positive tubes, N is the volume in the negative tubes (mL), and T is the total volume in *all* the tubes (mL). It is based on an assumption of random sampling from a Poisson distribution. For example, consider the dilution series 5x10 mL, 5x1 mL, 5x0.1 mL, and 5x0.01 mL where a pattern of 5-1-1-0 positives is obtained (as reported by Haas & Heller 1988). Then (14) gives MPN = 100 x 7/√(4.45 x 55.55) = 44.52 per 100 mL. Direct enumeration of the exact equations discussed above gives the result 43.20 per 100 mL (this result is also presented in Table 8).

Doing the calculations

A FORTRAN computer program (‘**XactMPN**’) has been written to do the appropriate exact calculations (computational details are given in Appendix A along with a listing of the code in Appendix B). Considerable care has been taken to make the code both accurate and as fast as possible, by using the direct algorithm (Equation 9) as much as possible. It also employs some transformations and back-transformations to minimise rounding errors when dealing with products and ratios of very large and very small numbers. It has been checked against the results presented for a 1x5x5 table by Tillett & Coleman (1985) and for a 5x5x5 table by Tillett (1987). Nevertheless it can be *very* slow in some extreme cases (e.g., for a 5-5-5-4 case in a 5x5x5x5 setup), and this needs further investigation.

Calculated MPN tables

1x3x3 *Campylobacter* table

We have $I = 1$, $J = K = 3$; $V_I = 500$ mL, $V_J = 100$ mL, $V_K = 10$ mL. Table 1 lists the probabilities of getting i, j, and k positive tubes for various ranges of numbers of bacteria (n) in this 1x3x3 series of total filtered volume 830 mL. So, for example, if the pattern of positive tubes was 1-2-0, then reading down the middle of the third column of Table 1 we find that the probability of observing this pattern if there were in fact $n = 6$ bacteria in the filtered volume is 0.36612; for 7 bacteria it is 0.37773 and for 8 bacteria it is 0.37241. That makes 7 bacteria (per 830 mL) the value of n at the mode of the probability distribution, and so it corresponds to the MPN.[4] Table 1 also gives the ‘occurrence probability’, being

---

[4] A check on accuracy of the program is afforded by noting that the sums of probabilities for a given value of n over all possible values of i, j, and k should be unity. Checks show that this holds true for the computed values,

the probability that a particular set of positive results could arise for a given number of bacteria present in the sample.

Table 2 lists the most probable numbers per 100 mL, retaining two decimal places in the computed result (obtained by taking the number in Table 1 with the maximum occurrence probability and dividing it by 8.3). Also shown is the MPN per litre, rounded to the nearest whole number. Table 2 shows all possible cases, no matter how improbable (Table 1 shows that some combinations are most improbable — the too-frequent obtaining of such results may often be taken as a signal that some laboratory error has been made (Woodward 1957). One can expect that these results will tend to be slightly lower than those calculated using approximations based on Poisson theory (Tillett 1995). Such tendencies are borne out in most cases of *Campylobacter* enumeration reported to me by Andrew Ball (ESR), using a computer program supplied by D. J. Best (based on Best 1990). Note however that Thomas's simple formula (Equation 14) may give a rather different pattern of discrepancies (e.g., as shown on the middle column of Table 2, in which the approximation gives results that are too high at low concentrations and vice versa).

Table 3 shows the MPNs calculated using the **XactMPN** program for a 1x3x3 series using subsample volumes chosen to fully utilise the 1 L sample taken (1x670 mL, 3x100 mL, 3x10 mL). Results differ but a little from those in Table 2. Further use of the software shows that the most crucial feature of the technique is the volume of the smallest subsample, rather than the total volume used. For example, using 1x500 mL, 3x100 mL and 3x5 mL, the 1-3-1 combination of positive tubes corresponds to an MPN of 8.10 per 100 mL (Table 4), whereas the same combination in the original series (1x500 mL, 3x100 mL, 3x10 mL, as on Table 2) gives an MPN of 4.46 /100 mL.

3x3x3 table

Table 5 shows the 3x3x3 table calculated using the **XactMPN** and **Greenwood_Yule_MPN** programs (listed in Appendices B and D). For comparison we show approximate MPNs reported by de Man (1983) and by Beliaeff & Mary (1993). The Beliaeff & Mary results agree (within rounding error) with the **Greenwood_Yule_MPN** result, as expected. These, and those of de Man (1983), are generally a little higher than the exact result. Also shown are the occurrence probabilities attached to the exact MPN values.

1x5x5 table

Table 6 shows a 1x5x5 table calculated using the **XactMPN** program. Also shown are MPNs reported in WHO's drinking-water guidelines, and those reported by Tillett & Coleman (1985), using the same exact method as in this report but rounded to the nearest integer. The WHO values usually differ by 1 or 2 units from the exact values.

Comparisons with 5x5x5 tables in Standard Methods

Table 7 shows comparisons between exact results and the approximate values reported in standard methods, i.e., the identical values in APHA's recommended seawater and shellfish procedures (APHA 1970) and their 14th edition of 'Standard methods ...' (APHA 1975), versus their different (and unexplained) values in the 19th edition of 'Standard methods ...' (APHA 1995). Also shown are exact results obtained by Tillett (1987), which she rounded to the nearest multiple of 5 for values above 50 and to the nearest 10 for values above 150.

---

e.g., Table 1 shows it to be true for $n = 2$ (the table can't be used to demonstrate this for higher n, because not all the required probabilities are shown).

5x5x5x5 table

Table 8 shows comparisons between exact results and the approximate values reported by Haas & Heller (1988), who also reported the frequency of obtaining various combinations of positives for this setup for 1,335 samples of Lake Michigan Chicago intake water from 1978 to 1980. This table is presented as laboratories often use a four-series dilution, yet standard tables seldom exist for this case. Furthermore, laboratories often find that in doing so either the first two sets of replicates are all positive, or the last two are all negative (John Mills, Agresearch, Hamilton, pers. comm.) and so use a three-series MPN table. For example, consider a setup consisting of 5x100 mL, 5x10 mL, 5x1 mL and 5x0.1 mL. If a 5-5-1-0 set of positive tubes is obtained, a common convention (Woodward 1957) is to use a three-series table to read the value for a 5-1-0 pattern (in a 5x10 mL, 5x1 mL, 5x0.1 mL series), or a 5-5-1 pattern (in a 5x100 mL, 5x10 mL, 5x1 mL series) and to take the value as the correct result. In fact this procedure is an approximation to the correct result, and may not be a very good one, because it ignores the volume of sample discarded. For example, the exact MPN for the 5-5-1-0 pattern is 32.76 per 100 mL (i.e., 182 bacteria in the total volume of 555.5 mL, cf. Table 8), whereas the 5-5-1 pattern gives an MPN of 34.59 per 100 mL and the 5-1-0 pattern gives an MPN of 30.63 per 100 mL (cf. Table 7).

Calculated 49x48 well IDEXX Quanti-Tray/2000™ tables

Tables 9 and 10 are based on a rapid new technique in a kit supplied by IDEXX Laboratories, Inc., Maine, USA, in which one presses a plastic bag containing 100 mL of sample and reagents into a tray containing 49 large wells and 48 small wells.[5] This tray is incubated for 24 hours and positive wells are noted by a colour change. This is used for enumeration of either *E. coli* (the 'Colilert' kit) or enterococci (the 'Enterolert' kit), via an approximate MPN table supplied with the kits.[6]

Tables 9 and 10 present calculated exact MPNs for this configuration of wells, using two constellations of well volumes: Table 9 uses large and small well volumes of 1.86 mL and 0.186 ml respectively, while Table 10 uses volumes of 1.88 mL and 0.16 mL. In both cases the total volume is about 100 mL (so that the total volume is 100.068 mL and 99.80 mL, respectively). The former is our best guess as to the actual volumes used by IDEXX to calculate the values in their MPN table (as advised by the New Zealand supplier, R. Fullerton, Environmental Diagnostics, Auckland). The latter was our first guess as to the volumes used.

The occurrence probabilities are indicted by the shaded regions in the tables, showing that well over half of the values in the table are highly improbable (i.e., with occurrence probability less than 10^{-6}).[7] These results are generally within 1–2% of the values in the IDEXX table (as shown in the contour graphs on Figures 1 and 2). However, much larger discrepancies can occasionally occur for plausible MPN values, as shown on those figures.

Comparison with the 51-well IDEXX Quanti-Tray™ table

Table 11 presents a comparison between exact results and the published tables for this configuration (also supplied to me by R. Fullerton, Environmental Diagnostics, Auckland).

---

[5] Actually one of the large wells is considerably larger than the rest, serving to accommodate any residual sample volume greater than 100 mL. This has been ignored in this report, as it also appears to have been in the development of the standard tables accompanying the IDEXX kit.

[6] Neither the calculation method nor the actual volumes used in the calculation have been sighted.

[7] In which case one has a 'possible but highly improbable Most Probable Number'.

Discussion

Nearly all published tables are calculated from approximate methods. The exceptions are the 1x5x5 of Tillett & Coleman (1985) and the 5x5x5 of Tillett (1987), which are exact. The latter table has been recommended for use in the UK (Public Health Laboratory Service Working Group 1992).

The unfortunate consequence of this proliferation is discrepancies between tables — made worse by the lack of explanation of the existence of, and reasons for, such differences. Furthermore, not all discrepancies are minor, e.g., in the 5x5x5 table of APHA (1970, 1975) the MPN for the 5-5-1 series of positives is given as 350 /100 mL, whereas APHA (1995) gives it as 300 /100 mL.[8] Table 7 shows the exact result to be 346 /100 mL.

This situation may be worsened by the promulgation of software for calculating tables for any dilution series, each producing slightly different answers (e.g., Parnow 1972, Hurley & Roscoe 1983, Best's program, based on Best 1990, Koch 1994). Hurley & Ruscoe's BASIC program, and a further BASIC procedure based on de Man (1975), are repeated in a draft international standard (ISO 1988).

In any attempt to provide better harmony between 'standard' tables, some agreement would also be needed on the appropriate rounding conventions, and on the actual volumes used in modern kits (the multi-well MPN results are particularly sensitive to the assignment of these volumes, as shown in Figures 1 and 2).

Optimising computer code to handle computationally demanding cases (especially 5-5-5-x patterns in a 5x5x5x5 setup) requires further investigation.

Confidence and credibility intervals

This section provides background for the material reported in McBride et al. (in press).

Although early theoretical developments focused on calculation of the MPN value itself (McCrady 1915, 1918, Greenwood & Yule 1917), attention has since been paid to quantify some measure of uncertainty about that value. Routinely, one uses 'confidence intervals', in which probabilities of various MPNs are computed for a range of assumed true concentrations: unlikely MPNs then fall into the tail of the distribution of those probabilities and so define the confidence limits. This is the 'relative frequency' approach — the probabilities refer to a *proportion of outcomes* under a particular hypothesis (the assumed concentrations). An example is the much-quoted paper of Woodward (1957), using an MPN-ordering procedure (as explained by Loyer & Hamilton 1984), results for which have been used in standard works (e.g., APHA 1975, WHO 1984). Different sets of intervals have been calculated using pattern-ordering in 'Sterne-type intervals' (Loyer & Hamilton 1984), and other lognormal approximations (Eisenhart & Wilson 1943, Cochran 1950, Best 1990). It has been held that Woodward's is the most accurate of these methods, especially when accompanied by necessary corrections (Loyer & Hamilton 1984). More recently, the US Food and Drug Administration has endorsed a narrower set of confidence intervals, being those published by de Man (1983) as discussed below.

However, there is a substantive and seldom-addressed issue with respect to the interpretation of confidence intervals. That is, because they are based on concepts of relative frequency, the 95% (or 99%) probability they invoke refers to the proportion of time that the interval would contain the true value, *were repeated assays to be performed.* But commonly only *one* assay (or a limited number) is performed and the analyst wishes to claim 95% probability that the sample assayed had a microbial

---

[8] This may be the result of rounding to the nearest hundred in later editions of 'Standard methods ...'.

concentration in the numerical range given by the confidence limits for the observed pattern of positive results. In that case the frequency interpretation has no meaning (Casella & Berger 1990), and one is actually making a Bayesian statement, i.e., a probability statement about the concentration given the pattern of positives obtained in the test. Furthermore, Bayesian probability calculations can only proceed if one invokes (even unwittingly) a 'prior probability distribution', that is, the analyst's view as to the distribution of concentration before performing the test. This is used, along with the data via a likelihood function, to calculate the posterior probability distribution (mimicking the learning process of updating previous views in the light of new information).

The specification of prior probability is a matter of some contention, because it appears to introduce subjectivity. Yet, if probability limits are desired for a particular result one *must* invoke a prior distribution — in order for it to be updated via Bayes' rule. In that regard note the views of two experienced environmental professionals, 'It is interesting that most researchers are taught statistics from a classical perspective, yet confidence intervals are often interpreted in a Bayesian sense. When the Bayesian interpretation is adopted, the analyst should realize that this implies a subjective interpretation for probability, and this should be specified in the analysis ... the prior probability distribution must be stipulated if the Bayesian interpretation for confidence intervals is adopted...' (Reckhow & Chapra 1983). Such intervals are more appropriately called 'credible intervals' (Lee 1997).

In fact, the early development of MPN theory was set in a Bayesian framework (McCrady 1915, 1918, Greenwood & Yule 1917), in which a diffuse uniform prior was adopted, stating that any positive concentration value is equally likely. (The manner in which this analysis leads to credible intervals was not pursued, because of its computational complexity.) More recently, the development of intervals put forward by de Man (1977) is equivalent to a Bayesian approach using a diffuse prior (Aspinall & Kilsby 1979, Loyer & Hamilton 1984), as the author later acknowledged (de Man 1983). Technically these are 'likelihood intervals' (Royall 1997) but in the MPN context they are equivalent to a credible interval with a diffuse prior. Interestingly, these results appear in many recent standard tables for food or water examination (e.g., APHA 1995, 2001) and so users of them are in fact using Bayesian intervals correctly. Whether that use is *appropriate* depends entirely on how reasonable the adopted prior distribution is. In at least some cases one can argue that it is not. For example, the diffuse prior posits that all concentrations are equally likely. This implies that a water body is more likely to be grossly contaminated than it is to be healthy (there being a much larger range of concentrations implying contamination), even when historical sampling has routinely demonstrated a healthy state. One can adopt other more 'informative' priors or adopt the 'Empirical Bayes' approach (Carlin & Louis 2000), in which the data are used to guide the choice and parameter(s) of the prior distribution. One such approach is to adopt a Poisson prior, based on the notion that microbes are distributed following a Poisson random process in the sampled environment (Broman et al. 1998), and using the calculated MPN as the mean of that distribution. A detailed description was given by Dalgety (1999), and is summarised in Appendix C.

Dalgety's approach is a 'naïve' Empirical Bayes method and so produces results that are 'overconfident' (Carlin & Louis 2000), i.e., his intervals are too short. This is because such methods 'use the data twice' (in the prior distribution and in the data likelihood function). This naïvety can be addressed by explicitly incorporating posterior uncertainty about the Poisson parameter (Carlin & Louis 2000); this is a fruitful research area.

Table 12 gives selected confidence and credible intervals for a 3x3x3 setup. The results shown for Beliaeff & Mary (1993 — reference **D** in the Table) are almost identical with those calculated from our implementation of the Greenwood & Yule approach (using the program **Greenwood_Yule_MPN** — see Appendix C).

Discussion

Confidence intervals

From Table 12 we see that the two 95% confidence interval results displayed are reasonably similar in their widths and limits although, because of their particular method of construction, the intervals of de Man (1983) are always shorter than those of Woodward (1957). If confidence intervals are to be used it therefore seems appropriate to endorse the intervals presented by de Man, especially as they have been incorporated into the U.S. Food & Drug Administration's (2001) 'Bacteriological Analytical Manual'.[9] A previous endorsement of Woodward's confidence intervals (Loyer & Hamilton 1984) was made before de Man's paper was published.

Credible intervals

The first 95% credible interval shown in Table 12, that of de Man (1977), is widely used (as noted above). As expected, it is very similar to the second and third credible intervals in that Table, developed using a diffuse prior (Greenwood & Yule 1917, Beliaeff & Mary 1993). These intervals are designed to have an area of 0.025 *in each tail* of their posterior distributions. These central credible intervals are also rather similar to the confidence intervals.

The fourth and fifth credible intervals on Table 12, the 'noncentral' case, have been obtained using a diffuse prior but requiring only that the *total tail area* is 0.05. These are guaranteed to be the shortest intervals satisfying this criterion and delimit the HPDR (Highest Posterior Density Region). Such regions have the added attraction that the probability density at any point inside the interval is greater than at any outside point (Lee 1997). Because the posterior MPN distribution is skewed to the right, both of the HPDR limits are always to the left of their central credible interval counterparts. As the results show, they are indeed always the shorter of the two types of credible intervals shown on the table. Almost identical results have been calculated using the equivalent procedures reported by Roussanov et al. (1996) using associated software.[10]

The last column of Table 12 shows the Poisson Empirical Bayes credible interval (Dalgety 1999), which is very much shorter than the others, reflecting the strong influence of the Poisson prior distribution on the results. Broman et al. (1998) have adopted a similar approach. Also, for this interval the central and HPDR intervals are very similar, because the posterior distribution with a Poisson prior tends to be quite symmetrical. As discussed in Appendix C, these intervals should possibly be lengthened a little to remove their 'naïve' component, but it is very likely that they will remain much shorter than the other intervals.

Some have proposed the use of the 'Most Probable Range' to quantify uncertainty, being the range of values with occurrence probabilities at least 95% of that for the MPN (Tillett 1987), though its arbitrariness has been noted (Beliaeff 1995). This term has also been used to refer to equi-tailed credible intervals (Dalgety 1999), for which perhaps a better term is MCR (Most Credible Range).

Finally, there is a question about what MPN estimate should be reported with credible intervals. It could be argued that when using a Bayesian credible interval approach the MPN should be read as the median or mean of the posterior distribution. My view is that it is better to use the exact value obtained from occupancy theory, because it is exact and because it is the least dependent on assumptions. This means that the program **XactMPN** is used to calculate the MPN, but the program **Greenwood_Yule_MPN** is used to calculate the credible intervals.[11]

---

[9] http://vm.cfsan.fda.gov/~ebam/bam-a2.html

[10] http://lib.stat.cmu.edu/general/bystub

[11] It may be appropriate to fuse these programs together at some later date and provide all MPN and interval options in the one (rather large) program.

Summary

While issues to do with MPN enumeration procedures can be resolved, by using exact methods, there is no one exact way to calculate measures of uncertainty about the MPN value. Most importantly, one must decide between the use of classical confidence intervals and Bayesian credible intervals. If the former is appropriate (i.e., the analyst wants to make probability statements about *performance in the long run*) then the approach of de Man (1983) should be used. If the Bayesian approach is to be taken (i.e., the analyst wants to make statements about *the current result*), it seems best to use noncentral intervals (i.e., HPDR), as given for two common setups by Beliaeff & Mary (1993). Software developed herein (implementing the Greenwood & Yule theory) gives essentially the same answers for those setups and can be applied to any other, but note that this approach assumes as its necessary prior distribution that gross contamination rather more likely than cleanliness. If that is not an appropriate assumption (and there seems to be many cases in which it will not be), then a Poisson empirical Bayes procedure may be employed. In so doing, the calculated uncertainty interval is much narrower.

Conclusions

The practice of developing standard tables from various approximate procedures should now be abandoned, because the result can be calculated exactly. The promulgation of computer code implementing these approximate procedures (e.g., Parnow 1972, Hurley & Roscoe 1983) therefore seems inappropriate.

Rounding conventions used in standard tables should always be stated. No particular convention is recommended here, other than to note that rounding a figure of 540 MPN /100 mL to 500 MPN /100 mL for a 5-5-2 pattern in a decimal dilution series (as noted in the Introduction) seems excessive: from Table 7 the exact integer value in that case is actually 541 MPN /100 mL.

In contrast, there is no one exact way to calculate measures of uncertainty about the MPN value. Most importantly, one must decide between the use of classical confidence intervals and Bayesian credible intervals. If the former are appropriate (i.e., the analyst wants to make probability statements *about performance in the long run*) then the intervals presented by de Man (1983) should be used. If the Bayesian approach is to be taken (i.e., the analyst wants to make statements *about the current result*), I recommend the use of noncentral intervals (i.e., HPDR), as given for two common setups by Beliaeff & Mary (1993). I favour use of the HPDR, rather than equi-tailed intervals, because they are the narrowest of all possible credible intervals and all probability densities inside the HPDR are greater than at any outside point. But note that this approach assumes as its necessary prior distribution that all concentrations are equally likely, implying that before obtaining new data the analyst held that a water body was more likely to be grossly contaminated than it was to be healthy, even when historical sampling had routinely demonstrated a healthy state. If this precautionary approach is not an appropriate assumption (and I can see many cases in which it will not be), then a Poisson Empirical Bayes procedure may be employed. In so doing the calculated uncertainty interval is much narrower. Further research is desirable on optimal forms of such intervals.

Acknowledgments

Partial funding for the preparation of this report was obtained through the Ministry of Health (contact: Drs M.E.U. Taylor and A. Kouzminov). Earlier work on the substance of the material presented in this report was partially funded through the Freshwater Microbiological Research Programme (administered by the Ministry for the Environment, funded through the Ministry of Research, Science and Technology). Also, some of the ideas and procedures have been worked out while considering associated issues under NIWA's FRST contracts C01819 ('Improved statistical methods for environmental issues') and C01X0215 ('Sustainability of Aquatic Ecosystems and Water Resources'). Helpful reviews were obtained from Rob Fullerton, Andrew Ball, and (especially) Chris Francis.

References

Abramowitz, M.; Stegun, I.A. (1972). Handbook of mathematical functions. 9th printing. Dover, New York. 1043 p.

Anon. (1983). Department of the Environment (UK). Department of Health and Social Security. Public Health Laboratory Service. The bacteriological examination of drinking water supplies. HMSO, London.

APHA (1970). Recommended procedures for the examination of sea water and shellfish. American Public Health Association, Washington D.C.

APHA (1975). Standard methods for the examination of water and wastewater. 14th ed. American Public Health Association, American Water Works Association, Water Pollution Control Federation, Washington D.C.

APHA (1995). Standard methods for the examination of water and wastewater. 19th ed. American Public Health Association, American Water Works Association, Water Environment Federation, Washington D.C.

APHA (2001). Compendium of methods for the microbiological examination of foods. 4th ed. American Public Health Association, Washington D.C.

Aspinall, L.J.; Kilsby, D.C. (1979). A microbiological quality control procedure based on tube counts. *Journal of Applied Bacteriology 46*: 325–329.

Beliaeff, B.; Mary, J.-Y. (1993). The "most probable number" estimate and its confidence limits. *Water Research 27(5)*: 799–805.

Beliaeff, B. (1995). Reply to Tillett's discussion. *Water Research 29(4)*: 1215.

Best, D.J. (1990). Optimal determination of most probable numbers. *International Journal of Food Microbiology 11*: 159–166.

Broman, K.: Speed, T.; Tigges, M. (1998). Estimation of antigen-responsive T Cell frequencies in PBMC from human subjects. *Statistical Science 13*: 4–8.

Carlin, B.P.; Louis, T.A. (2000). Bayes and empirical Bayes methods for data analysis. Chapman & Hall/CRC, Boca Raton, LA.

Casella, G.; Berger, R.L. (1990). Statistical inference. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Cochran, W.G. (1950). Estimation of bacterial densities by means of the "Most Probable Number". *Biometrics 6*: 105–116.

Dalgety, M.H. (1999). Establishing a Most Probable Range using existing Most Probable Number estimation techniques. BSc(Hons) project report 0655.420, Department of Statistics, University of Waikato, Hamilton, New Zealand. 53 p.

David, F.N.; Barton, D.E. (1962). Combinatorial chance. Griffin, London. 356 p.

de Man, J.C. (1975). The probability of most probable numbers. *European Journal of Applied Microbiology 1*: 67–78.

de Man, J.C. (1977). MPN tables for more than one test. *European Journal of Applied Microbiology 4*: 307–316.

de Man, J.C. (1983). MPN tables, corrected. *European Journal of Applied Microbiology 17*: 301–305.

Dorn, W.S.; McCracken, D.D. (1972). Numerical methods with Fortran IV case studies. Wiley, New York.

Eisenhart, C.; Wilson, P.W. (1943). Statistical methods and control in bacteriology. *Bacteriological Reviews 7*: 57–137.

Freund, J.E. (1992). Mathematical statistics. 5th ed. Prentice Hall, Upper Saddle River, NJ.

Greenwood, M.; Jnr.; Yule, G.U. (1917). On the statistical interpretation of some bacteriological methods employed in water analysis. *Journal of Hygiene 16*: 36–54.

Haas, C.N.; Heller, B. (1988). Test of the validity of the Poisson assumption for analysis of Most-Probable-Number results. *Applied and Environmental Microbiology 54(12)*: 2996–3002.

Hurley, M.A.; Roscoe, M.E. (1983). Automated statistical analysis of microbial enumeration by dilution series. *Journal of Applied Bacteriology 55*: 159–164.

ISO (1998). Water quality — Detection and enumeration of intestinal enterococci in surface and waste water — Part 1: Miniaturized method (Most Probable Number) by inoculation in liquid medium. Final draft international standard: ISO/FDIS 7899-1988(E). International Standards Organisation, New York.

Koch, A.L. (1994). Growth measurement. *In*: Methods for general and molecular bacteriology. Gerhardt, P.; Murray, R.G.E.; Wood, W.A.; Krieg, N.R. (eds). American Society for Microbiology, Washington D.C. pp. 257–260.

Lanczos, C. (1964). A precision approximation of the gamma function. *SIAM Journal on Numerical Analysis, Series B 1*: 86–96.

Lee, P.M. (1997). Bayesian statistics. an introduction. 2nd ed. Arnold, London.

Loyer, M.W.; Hamilton, M.A. (1984). Interval estimation of the density of organisms using a serial-dilution experiment. *Biometrics 40*: 907–916.

McBride, G.B.; Till, D.; Ryan, T.; Ball, A. Lewis, G.; Palmer, S.; Weinstein, P. (2002). Freshwater Microbiology Research Programme. Pathogen occurrence and human health risk assessment analysis. Ministry for the Environment Technical Publication. 93 p.[12]

McBride, G.B.; McWhirter, J. L.; Dalgety, M.H. (in press). Accounting for uncertainty in Most Probable Number enumerations for microbiological assays. *JAOAC (Journal of the Association of Analytical Communities)*.

McCrady, M.H. (1915). The numerical interpretation of fermentation-tube results. *Journal of Infectious Diseases 17*: 183–212.

McCrady, M.H. (1918). Tables for rapid interpretation of fermentation-tube results. *The Public Health Journal (Canada) 9(5)*: 201–220.

Parnow, R.J. (1972). Computer program estimates bacterial densities by means of most probable numbers. *Food Technology 26*: 56–62.

Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. (1986). Numerical recipes in FORTRAN. The art of scientific computing. Cambridge University Press, Cambridge, UK.

Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. (1992). Numerical recipes in FORTRAN. The art of scientific computing 2nd ed. Cambridge University Press, Cambridge UK.

---

[12] Available at http://www.mfe.govt.nz/publications/water/freshwater-microbiology-nov02

Public Health Laboratory Service Working Group (1992). Bacteriological examination of shellfish. Ministry of Agriculture, Fisheries and Food, Department of Health (UK). *PHLS Microbiology Digest 9(2)*: 76–82.

Reckhow, K.H.; Chapra, S.C. (1983). Engineering approaches for lake management, volume 1: data analysis and empirical modeling. Butterworth, Boston.

Roberts, F.S. (1984). Applied combinatorics. Prentice-Hall, Englewood Cliffs, NJ.

Roussanov, B.; Hawkins, D.M.; Tatini, S.R. (1996). Estimating bacterial density from tube dilution data by a Bayesian method. *Food Microbiology 13(5)*: 341–363.

Royall, R. (1997). Statistical evidence: a likelihood paradigm. Chapman & Hall/CRC, Boca Raton, LA.

Swaroop, S. (1938). Numerical estimation of *B. coli* by dilution method. *Indian Journal of Medical Research 26*: 353–377.

Thomas, H.A. (1942). Bacterial densities from fermentation tube tests. *Journal of the American Water Works Association 34*: 572–576.

Tillett, H.E. (1987). Most probable number of organisms: revised tables for the multiple tube method. *Epidemiology and Infection 99*: 471–476.

Tillett, H.E. (1995). Comment on "The 'most probable number' estimate and its confidence limits." *Water Research 29(4)*: 1213–1214.

Tillett, H.E.; Coleman, R. (1985). Estimated numbers of bacteria from non-homogeneous bodies of water. *Journal of Applied Bacteriology 59*: 381–388.

von Mises, R. (1942). On the correct use of Bayes' formula. *Annals of Mathematical Statistics 13*: 156–165.

WHO (1984). Guidelines for drinking-water quality. World Health Organization, Geneva.

Wolfram, S. (1992). Mathematica: a system for doing mathematics by computer. 2nd ed. Addison Wesley, Reading, Mass.

Woodward, R.L. (1957). How probable is the Most Probable Number? *Journal of the American Water Works Association 49*: 1060–1068.

Table 1: Occurrence probabilities ('Prob') for possible numbers ('*n*') of bacteria in a 1x500 mL, 3x100 mL, 3x10 mL series of tubes using the program XactMPN.

n	Prob
0-0-1	
1	0.03614
2	0.00044
0-0-2	
2	0.00087
3	0.00003
0-0-3	
3	0.00001
4	0.00000
0-1-0	
1	0.36145
2	0.04355
0-1-1	
2	0.02613
3	0.00519
0-1-2	
3	0.00094
4	0.00027
0-1-3	
4	0.00002
5	0.00001
0-2-0	
2	0.08710
3	0.03148
0-2-1	
3	0.00944
4	0.00478
0-2-2	
4	0.00046
5	0.00030
0-2-3	
5	0.00001
6	0.00001
0-3-0	
3	0.01049
4	0.00759
0-3-1	
4	0.00152
5	0.00142
6	0.00088
0-3-2	
5	0.00009
6	0.00011
7	0.00008
0-3-3	
6	0.00000
7	0.00000
8	0.00000
1-0-0	
1	0.60241
2	0.36290
1-0-1	
2	0.04355
3	0.04014
4	0.03257
1-0-2	
3	0.00157
4	0.00197
5	0.00202
6	0.00186
7	0.00160
1-0-3	
4	0.00003
5	0.00004
6	0.00005
7	0.00005
8	0.00005
9	0.00005
1-1-0	
2	0.43548
3	0.47220
4	0.42353
1-1-1	
3	0.04722
4	0.06941
5	0.07862
6	0.07962
7	0.07536
8	0.06809
1-1-2	
4	0.00228
5	0.00425
6	0.00584
7	0.00696
8	0.00759
9	0.00779
10	0.00763
11	0.00721
12	0.00662
1-1-3	
5	0.00005
6	0.00010
7	0.00017
8	0.00023
9	0.00029
10	0.00033
11	0.00036
12	0.00037
13	0.00037
14	0.00036
15	0.00035
16	0.00032
1-2-0	
3	0.15740
4	0.26550
5	0.33130
6	0.36612
7	0.37773
8	0.37241
9	0.35547
10	0.33122
1-2-1	
4	0.02276
5	0.04867
6	0.07360
7	0.09571
8	0.11379
9	0.12723
10	0.13603
11	0.14053
12	0.14135
13	0.13915
14	0.13460
15	0.12835
16	0.12092
1-2-2	
5	0.00137
6	0.00357
7	0.00636
8	0.00953
9	0.01285
10	0.01609
11	0.01908
12	0.02167
13	0.02380
14	0.02543
15	0.02656
16	0.02723
17	0.02747
18	0.02734
19	0.02689
20	0.02618
21	0.02527
22	0.02420
1-2-3	
6	0.00003
7	0.00010
8	0.00021
9	0.00036
10	0.00054
11	0.00075
12	0.00097
13	0.00121
14	0.00144
15	0.00166
16	0.00186
17	0.00204
18	0.00220
19	0.00233
20	0.00243
21	0.00250
22	0.00255
23	0.00257
24	0.00257
25	0.00255
26	0.00252
27	0.00246
28	0.00240
29	0.00232
30	0.00224
1-3-0	
4	0.02529
5	0.06093
6	0.10094
—	—
17	0.38094
18	0.38430
19	0.38519
20	0.38398
—	—
26	0.34885
27	0.34039
1-3-1	
5	0.00457
6	0.01338
7	0.02610
—	—
36	0.43660
37	0.43700
38	0.43689
39	0.43631
—	—
53	0.39600
54	0.39173
1-3-2	
6	0.00033
7	0.00114
8	0.00257
—	—
87	0.44825
88	0.44857
89	0.44879
90	0.44890
91	0.44891
92	0.44882
93	0.44864
94	0.44836
95	0.44798
96	0.44752
—	—
125	0.40496
126	0.40278
1-3-3	
10	0.00020
—	—
99	0.33846
100	0.33919
—	—
∞	1.00000

'Prob' is the occurrence probability, of getting the stated (italicised) pattern of positives in the 1x3x3 set of tubes, given the number of bacteria shown under '*n*'. For example, at the bottom of the middle column the probability of getting a 1-2-1 pattern of positive results if there were in fact a total of 16 bacteria present is 0.12092. In terms of the symbols used in the theory section this is $\Pr(i, j, k \mid n) = \Pr(1, 2, 1 \mid 16) = 0.12092$. For this combination of positive tubes the probability mode is at $n = 12$ (where 'Prob' = 0.14135), so the MPN = 12/8.3 = 1.45 /100 mL. (Calculation of these probabilities was halted when they had decreased to less than 90% of the maximum.)

Table 2: **Most Probable Numbers of bacteria for all possible combinations of tubes showing a positive reaction in the series 1x500 mL, 3x100 mL, 3x10 mL using the program XactMPN.**

Combination of positives	**MPN /100 mL**[†]		**MPN /L**[‡]
0-0-0	0.00	(0.00)	<1
0-0-1	0.12	(0.12)	1
0-0-2	0.24	(0.24)	2
0-0-3	0.36	(0.37)	4
0-1-0	0.12	(0.13)	1
0-1-1	0.24	(0.26)	2
0-1-2	0.36	(0.39)	4
0-1-3	0.48	(0.52)	5
0-2-0	0.24	(0.28)	2
0-2-1	0.36	(0.42)	4
0-2-2	0.48	(0.56)	5
0-2-3	0.60	(0.71)	6
0-3-0	0.36	(0.45)	4
0-3-1	0.48	(0.61)	5
0-3-2	0.72	(0.77)	7
0-3-3	0.84	(0.93)	8
1-0-0	0.12	(0.19)	1
1-0-1	0.24	(0.39)	2
1-0-2	0.60	(0.59)	6
1-0-3	0.84	(0.80)	8
1-1-0	0.36	(0.46)	4
1-1-1	0.72	(0.70)	7
1-1-2	1.08	(0.96)	11
1-1-3	1.57	(1.23)	16
1-2-0	0.84	(0.91)	8
1-2-1	1.45	(1.27)	14
1-2-2	2.05	(1.65)	20
1-2-3	2.89	(2.08)	29
1-3-0	2.29	(2.53)	23
1-3-1	4.46	(3.88)	45
1-3-2	10.96	(6.59)	110
1-3-3	>10.96	(>6.59)	>110

[†] Values in parentheses have been calculated from Thomas' approximation formula — equation 14.

[‡] Rounded to the nearest integer. The MPN for the 0-0-0 case can be stated as 0 per 100 mL because more than 100 mL of sample was used; but less than 1 L was used, so the equivalent result is <1 per L.

Table 3: **Most Probable Numbers of bacteria for all possible combinations of tubes showing a positive reaction in the series 1x670 mL, 3x100 mL, 3x10 mL using the program XactMPN.**

Combination of positives	**MPN /100 mL**	**MPN /L**[†]
0-0-0	0	0
0-0-1	0.1	1
0-0-2	0.2	2
0-0-3	0.3	3
0-1-0	0.1	1
0-1-1	0.2	2
0-1-2	0.3	3
0-1-3	0.4	4
0-2-0	0.2	2
0-2-1	0.3	3
0-2-2	0.4	4
0-2-3	0.5	5
0-3-0	0.3	3
0-3-1	0.4	4
0-3-2	0.5	5
0-3-3	0.7	7
1-0-0	0.1	1
1-0-1	0.3	3
1-0-2	0.6	6
1-0-3	0.9	9
1-1-0	0.3	3
1-1-1	0.7	7
1-1-2	1.1	11
1-1-3	1.5	15
1-2-0	0.9	9
1-2-1	1.4	14
1-2-2	2.1	21
1-2-3	2.8	28
1-3-0	2.3	23
1-3-1	4.5	45
1-3-2	10.9	109
1-3-3	>10.9	>109

[†] No rounding required (total sample examined = 1 L). The MPN for the 0-0-0 case can be stated as 0 per 100 mL and 0 per L because 1 L of sample was used.

Table 4: **Most Probable Numbers of bacteria for all possible combinations of tubes showing a positive reaction in the series 1x500 mL, 3x100 mL, 3x5 mL using the program XactMPN.**

Combination of positives	MPN /100 mL	MPN /L[†]
0-0-0	0	<1
0-0-1	0.12	1
0-0-2	0.25	2
0-0-3	0.37	4
0-1-0	0.12	1
0-1-1	0.25	2
0-1-2	0.37	4
0-1-3	0.49	5
0-2-0	0.25	2
0-2-1	0.37	4
0-2-2	0.49	5
0-2-3	0.61	6
0-3-0	0.37	4
0-3-1	0.49	5
0-3-2	0.74	7
0-3-3	0.86	9
1-0-0	0.12	1
1-0-1	0.25	2
1-0-2	0.61	6
1-0-3	0.86	9
1-1-0	0.37	4
1-1-1	0.74	7
1-1-2	1.10	11
1-1-3	1.60	16
1-2-0	0.98	10
1-2-1	1.47	15
1-2-2	2.21	22
1-2-3	2.94	29
1-3-0	2.94	29
1-3-1	8.10	81
1-3-2	21.84	218
1-3-3	>21.84	>218

[†] Rounded to the nearest integer ('0.25' is actually 2/8.15 = 0.2454 < 0.25, and so rounds to 2 /L). The MPN for the 0-0-0 case can be stated as 0 per 100 mL because more than 100 mL of sample was used; but less than 1 L was used, so the equivalent result is <1 per L.

Table 5: **Most Probable Numbers of bacteria (per 100 mL) for all possible combinations of tubes showing a positive reaction in the series 3x100 mL, 3x10 mL, 3x1 mL.**

Combination of positives	Greenwood & Yule (1917)[a]	Beliaeff & Mary (1993)	de Man (1983)	Using XactMPN	
				MPN	Occur Prob
0-0-0			<0.30	0	–
0-0-1	0.30	0.3	0.30	0.30	0.009
0-0-2	–	–	–	0.60	5x10^{-4}
0-0-3	–	–	–	0.90	2x10^{-7}
0-1-0	0.30	0.3	0.30	0.30	0.090
0-1-1	–	–	0.61	0.60	0.002
0-1-2	–	–	–	0.90	1x10^{-5}
0-1-3	–	–	–	1.20	6x10^{-8}
0-2-0	0.62	0.6	0.62	0.60	0.005
0-2-1	–	–	–	0.90	0.001
0-2-2	–	–	–	1.20	2x10^{-6}
0-2-3	–	–	–	1.50	9x10^{-9}
0-3-0	–	–	0.94	0.90	2x10^{-4}
0-3-1	–	–	–	1.20	6x10^{-6}
0-3-2	–	–	–	1.50	9x10^{-8}
0-3-3	–	–	–	1.80	5x10^{-10}
1-0-0	0.36	0.4	0.36	0.30	0.901
1-0-1	0.72	0.7	0.72	0.60	0.016
1-0-2	–	–	1.1	0.90	1x10^{-4}
1-0-3	–	–	–	1.20	6x10^{-7}
1-1-0	0.74	0.7	0.74	0.60	0.162
1-1-1	1.12	1.1	1.1	0.90	0.004
1-1-2	–	–	–	1.20	5x10^{-5}
1-1-3	–	–	–	1.80	3x10^{-7}
1-2-0	1.14	1.1	1.1	0.90	0.015
1-2-1	1.54	1.5	1.5	1.20	5x10^{-4}
1-2-2	–	–	–	1.80	9x10^{-6}
1-2-3	–	–	–	2.10	6x10^{-8}
1-3-0	1.57	1.6	1.6	1.20	6x10^{-4}
1-3-1	–	–	–	1.80	3x10^{-5}
1-3-2	–	–	–	2.10	7x10^{-7}
1-3-3	–	–	–	2.70	6x10^{-9}
2-0-0	0.92	0.9	0.92	0.60	0.541
2-0-1	1.43	1.4	1.4	1.20	0.018
2-0-2	–	–	2.0	1.80	3x10^{-4}
2-0-3			–	2.40	2x10^{-6}
2-1-0	1.47	1.5	1.5	1.20	0.184
2-1-1	2.05	2.0	2.0	1.80	0.009
2-1-2	–	–	2.7	2.40	2x10^{-4}
2-1-3			–	3.00	2x10^{-6}
2-2-0	2.11	2.1	2.1	1.80	0.033
2-2-1	2.76	2.8	2.8	2.40	0.002
2-2-2	–	–	3.5	3.30	7x10^{-5}
2-2-3	–	–	–	3.90	9x10^{-7}
2-3-0	2.86	2.9	2.9	2.70	0.003
2-3-1	–	–	3.6	3.30	3x10^{-4}
2-3-2	–	–	–	4.20	1x10^{-5}
2-3-3	–	–	–	5.11	2x10^{-7}
3-0-0	2.31	2.3	2.3	2.10	0.398
3-0-1	3.85	3.9	3.8	3.60	0.034
3-0-2	6.36	6.4	6.4	6.01	0.002
3-0-3	–	–	–	9.31	4x10^{-5}
3-1-0	4.27	4.3	4.3	3.90	0.400
3-1-1	7.49	7.5	7.5	7.21	0.069
3-1-2	11.52	12	12	11.41	0.007
3-1-3	–	–	16	15.62	3x10^{-4}
3-2-0	9.33	9.3	9.3	9.01	0.339
3-2-1	14.94	15	15	14.71	0.129
3-2-2	21.47	21	21	21.32	0.025
3-2-3	29.17	29	29	28.83	0.002
3-3-0	23.98	24	24	23.72	0.370
3-3-1	46.22	46	46	45.95	0.430
3-3-2	109.89	110	110	109.61	0.446
3-3-3	/	/	>110	>109.61	–

[a] Calculated using program **Greenwood_Yule_MPN**, described in Appendix C.

Table 6: **Most Probable Numbers of bacteria (/100 mL) for selected combinations of tubes showing a positive reaction in the series 1x50 mL, 5x10 mL and 5x1 mL.**

Combination of positives	WHO (1984)*	Tillett & Coleman (1985)	XactMPN
0-0-0	–	0	0
0-0-1	1	1	0.95
0-0-2	2	–	1.90
0-1-0	1	1	0.95
0-1-1	2	2	1.90
0-1-2	3	–	2.86
0-2-0	2	2	1.90
0-2-1	3	3	2.86
0-2-2	4	–	3.81
0-3-0	3	3	2.86
0-3-1	5	–	3.81
0-4-0	5	–	3.81
1-0-0	1	1	0.95
1-0-1	3	2	1.90
1-0-2	4	–	2.86
1-0-3	6	–	4.76
1-1-0	3	2	1.90
1-1-1	5	4	3.81
1-1-2	7	6	5.71
1-1-3	9	–	7.62
1-2-0	5	4	3.81
1-2-1	7	7	6.67
1-2-2	10	9	8.57
1-2-3	12	–	11.43
1-3-0	8	8	7.62
1-3-1	11	10	10.48
1-3-2	14	13	13.33
1-3-3	18	17	17.14
1-3-4	21	–	20.95
1-4-0	13	12	12.38
1-4-1	17	16	16.19
1-4-2	22	21	20.95
1-4-3	28	27	26.67
1-4-4	35	33	33.33
1-4-5	43	–	41.90
1-5-0	24	23	22.86
1-5-1	35	33	33.33
1-5-2	54	53	53.33
1-5-3	92	91	91.43
1-5-4	161	160	160.00
1-5-5	–	‘∞’	>160.00

* Based on Swaroop (1938)

Table 7: **Comparison of MPN (/100 mL) for 5x10 mL, 5x1 mL, 5x0.1 mL in standard tables and using this report's computer program.**

Combination of positives	APHA (1970, 1975)	APHA (1995)	Tillett (1987)	XactMPN
0-0-0	<2	<2	'none found in 55.5 ml'	<1.8
0-0-1	2	2	2	1.8
0-1-0	2	2	2	1.8
0-2-0	4	4	–	3.6
1-0-0	2	2	2	1.8
1-0-1	4	4	4	3.6
1-1-0	4	4	4	3.6
1-1-1	6	6	–	5.4
1-2-0	6	6	5	5.4
2-0-0	5	4	4	3.6
2-0-1	7	7	5	5.4
2-1-0	7	7	5	5.4
2-1-1	9	9	7	7.2
2-2-0	9	9	7	7.2
2-3-0	12	12	11	10.8
3-0-0	8	8	7	7.2
3-0-1	11	11	9	9.0
3-1-0	11	11	9	9.0
3-1-1	14	14	13	12.6
3-2-0	14	14	13	12.6
3-2-1	17	17	16	16.2
3-3-0	–	–	16	16.2
4-0-0	13	13	11	10.8
4-0-1	17	17	14	14.4
4-1-0	17	17	16	16.2
4-1-1	21	21	20	19.8
4-1-2	26	26	–	23.4
4-2-0	22	22	20	19.8
4-2-1	26	26	25	25.2
4-3-0	27	27	25	25.2
4-3-1	33	33	31	30.6
4-4-0	34	34	32	32.4
4-4-1	–	–	38	37.8
5-0-0	23	23	22	21.6
5-0-1	31	30	29	28.8
5-0-2	43	40	41	41.4
5-1-0	33	30	31	30.6
5-1-1	46	50	43	43.2
5-1-2	63	60	60	61.3
5-1-3	–	–	85	82.9
5-2-0	49	50	50	46.9
5-2-1	70	70	70	68.5
5-2-2	94	90	95	93.7
5-2-3	–	–	120	119
5-3-0	79	80	75	77.5
5-3-1	110	110	110	108
5-3-2	140	140	140	139
5-3-3	180	170	175	173
5-3-4	–	–	210	211
5-4-0	130	130	130	128
5-4-1	170	170	170	171
5-4-2	220	220	220	220
5-4-3	280	280	280	277
5-4-4	350	350	345	346
5-5-0	240	240	240	238
5-5-1	350	300	350	346
5-5-2	540	500	540	541
5-5-3	920	900	910	917
5-5-4	1600	1600	1600	1609
5-5-5	≥2400	≥1600	'1800+'	>1609

Table 8: **Most Probable Numbers of bacteria (/100 mL) for selected combinations of tubes showing a positive reaction in the series 5x10 mL, 5x1 mL, 5x0.1 mL, and 5x0.01 mL.**

Observed positives at vol (mL)				**Frequency**	**Haas & Heller (1998)**	**XactMPN**	**Occurrence probability**
10	**1**	**0.1**	**0.01**				
0	0	0	0	213	0.0	0.0	–
2	0	0	0	200	4.47	3.60	0.648
1	0	0	0	182	1.98	1.80	0.900
3	0	0	0	146	7.77	7.20	0.378
5	0	0	0	115	23.04	21.60	0.191
4	0	0	0	111	12.73	10.80	0.265
5	1	0	0	76	32.74	30.60	0.286
4	1	0	0	42	16.85	16.20	0.198
3	1	0	0	34	10.69	9.00	0.176
2	1	0	0	26	6.83	5.40	0.175
5	2	0	0	23	48.9	46.80	0.268
5	4	0	0	15	127.56	126.01	0.192
1	1	0	0	13	4.02	3.60	0.162
4	2	0	0	11	21.56	19.80	0.080
5	2	1	0	11	69.2	68.41	0.080
5	3	0	0	8	78.2	77.41	0.225
2	2	0	0	8	9.3	7.20	0.025
5	0	1	0	7	31.24	28.80	0.025
5	5	1	0	6	329.06	327.63	0.266
5	4	1	0	6	168.88	167.42	0.152
5	1	1	0	6	45.29	43.20	0.054
3	2	0	0	6	13.82	12.60	0.043
0	1	0	0	5	1.82	1.80	0.090
3	0	1	0	5	10.55	9.00	0.017
5	5	4	0	4	1299.3	1297.93	0.200
5	5	5	1	3	3476.7	*3459.60*	*0.315*
5	5	2	0	3	493.22	491.45	0.261
4	0	1	0	3	16.54	14.40	0.019
2	3	0	0	3	11.88	10.80	0.003
5	5	5	0	2	2397.9	2,396.02	0.187
5	5	0	0	2	231.16	228.62	0.169
5	3	1	0	2	108.6	106.21	0.108
5	3	2	0	2	138.42	136.81	0.028
4	3	0	0	2	27.01	25.20	0.022
2	0	1	0	2	6.77	5.40	0.018
1	0	1	0	2	3.99	3.60	0.016
5	1	0	1	2	44.99	43.20	0.016
4	0	2	0	2	20.69	19.80	0.007
3	2	1	0	2	16.94	16.20	0.003
5	0	1	1	2	42.15	39.60	4x10^{-4}

The data have been reordered within each frequency band, according to descending values of their occurrence probabilities. Occurrence probabilities were calculated using **XactMPN**. Italicised values are approximate (**XactMPN** calculations not yet completed). These were evaluated from a 5x1 mL, 5x0.1 mL and 5x0.01 mL series, omitting the first 10 mL result (in all cases the first two sets of tubes were all positive).

Table 8 (ctd.): **Most Probable Numbers of bacteria (/100 mL) for selected combinations of tubes showing a positive reaction in the series 5x10 mL, 5x1 mL, 5x0.1 mL, and 5x0.01 mL.**

Observed positives at vol (mL)				**Frequency**	**Haas & Heller (1998)**	**XactMPN**	**Occurrence probability**
10	**1**	**0.1**	**0.01**				
5	5	5	3	1	9178.4	*9171.17*	*0.346*
5	5	3	0	1	793.43	792.08	0.226
5	5	4	1	1	1723.8	1722.77	0.162
5	5	3	1	1	1086.4	1085.51	0.111
5	5	2	1	1	699.64	698.47	0.079
5	4	2	0	1	216.09	214.22	0.064
1	2	0	0	1	6.12	5.40	0.018
4	1	1	0	1	21.11	19.80	0.018
5	4	0	1	1	165.77	163.82	0.014
5	2	2	0	1	93.22	91.81	0.013
3	1	1	0	1	13.63	12.60	0.010
4	2	1	0	1	26.38	25.20	0.009
3	3	0	0	1	17.19	16.20	0.007
5	1	2	0	1	62.49	61.21	0.006
5	3	3	0	1	172.16	171.02	0.005
4	4	0	0	1	33.41	32.40	0.004
5	4	4	0	1	335.12	333.03	0.003
2	0	2	0	1	9.11	7.20	3x10^{-3}
4	4	2	0	1	46.48	45.00	6x10^{-5}
1	1	1	1	1	6.07	7.20	2x10^{-5}
4	5	2	0	1	55.71	54.01	7x10^{-6}
1	3	1	1	1	12.51	10.80	3x10^{-7}
4	3	4	0	1	51.87	50.41	2x10^{-7}
5	5	5	2	–	–	*5405.41*	*0.338*
5	5	5	4	–	–	*16090.09*	*0.410*

The data have been reordered within each frequency band, according to descending values of their occurrence probabilities. Occurrence probabilities were calculated using **XactMPN**. Italicised values are approximate (**XactMPN** calculations not yet completed). These were evaluated from a 5x1 mL, 5x0.1 mL and 5x0.01 mL series, omitting the first 10 mL result (in all cases the first two sets of tubes were all positive).

Table 9: Calculated MPN values for the 49x48 well IDEXX Quanti-Tray/2000™ (output from FORTRAN program Q4948).

MPNs for 49 large well x 48 small well QuantiTray.
Large well volume = 1.86 mL, small well volume = .186 mL.

# large wells positive	# small wells positive																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0
1	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.0
2	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	26.0	27.0
3	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	23.0	24.0	25.0	26.0	27.0	28.0
4	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	28.0	29.0
5	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	29.0	30.0	31.0
6	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.0	27.0	28.0	29.0	30.0	31.0	32.0
7	7.0	8.0	9.0	10.0	11.0	12.0	13.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	25.0	26.0	27.0	28.0	29.0	30.0	31.0	33.0	34.0
8	8.0	9.0	10.0	11.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	23.0	24.0	25.0	26.0	27.0	28.0	29.0	31.0	32.0	33.0	34.0	35.0
9	9.0	10.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	21.0	22.0	23.0	24.0	25.0	26.0	27.0	29.0	30.0	31.0	32.0	33.0	34.0	36.0	37.0
10	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	20.0	21.0	22.0	23.0	24.0	25.0	27.0	28.0	29.0	30.0	31.0	32.0	34.0	35.0	36.0	37.0	38.0
11	12.0	13.0	14.0	15.0	16.0	17.0	19.0	20.0	21.0	22.0	23.0	24.0	26.0	27.0	28.0	29.0	30.0	32.0	33.0	34.0	35.0	36.0	38.0	39.0	40.0
12	13.0	14.0	15.0	16.0	18.0	19.0	20.0	21.0	22.0	24.0	25.0	26.0	27.0	28.0	29.0	31.0	32.0	33.0	34.0	36.0	37.0	38.0	39.0	40.0	42.0
13	14.0	15.0	17.0	18.0	19.0	20.0	21.0	23.0	24.0	25.0	26.0	27.0	29.0	30.0	31.0	32.0	33.0	35.0	36.0	37.0	38.0	40.0	41.0	42.0	43.0
14	16.0	17.0	18.0	19.0	20.0	22.0	23.0	24.0	25.0	26.0	28.0	29.0	30.0	31.0	33.0	34.0	35.0	36.0	38.0	39.0	40.0	41.0	43.0	44.0	45.0
15	17.0	18.0	19.0	21.0	22.0	23.0	24.0	26.0	27.0	28.0	29.0	30.0	32.0	33.0	34.0	35.0	37.0	38.0	39.0	41.0	42.0	43.0	44.0	46.0	47.0
16	18.0	20.0	21.0	22.0	23.0	25.0	26.0	27.0	28.0	30.0	31.0	32.0	33.0	35.0	36.0	37.0	38.0	40.0	41.0	42.0	44.0	45.0	46.0	48.0	49.0
17	20.0	21.0	22.0	24.0	25.0	26.0	27.0	29.0	30.0	31.0	32.0	34.0	35.0	36.0	38.0	39.0	40.0	42.0	43.0	44.0	46.0	47.0	48.0	50.0	51.0
18	21.0	23.0	24.0	25.0	26.0	28.0	29.0	30.0	32.0	33.0	34.0	35.0	37.0	38.0	39.0	41.0	42.0	43.0	45.0	46.0	48.0	49.0	50.0	52.0	53.0
19	23.0	24.0	25.0	27.0	28.0	29.0	31.0	32.0	33.0	35.0	36.0	37.0	39.0	40.0	41.0	43.0	44.0	45.0	47.0	48.0	50.0	51.0	52.0	54.0	55.0
20	24.0	26.0	27.0	28.0	30.0	31.0	32.0	34.0	35.0	36.0	38.0	39.0	40.0	42.0	43.0	45.0	46.0	47.0	49.0	50.0	52.0	53.0	54.0	56.0	57.0
21	26.0	27.0	29.0	30.0	31.0	33.0	34.0	35.0	37.0	38.0	40.0	41.0	42.0	44.0	45.0	47.0	48.0	49.0	51.0	52.0	54.0	55.0	57.0	58.0	60.0
22	28.0	29.0	30.0	32.0	33.0	35.0	36.0	37.0	39.0	40.0	41.0	43.0	44.0	46.0	47.0	49.0	50.0	52.0	53.0	55.0	56.0	57.0	59.0	60.0	62.0
23	29.0	31.0	32.0	34.0	35.0	36.0	38.0	39.0	41.0	42.0	43.0	45.0	46.0	48.0	49.0	51.0	52.0	54.0	55.0	57.0	58.0	60.0	61.0	63.0	64.0
24	31.0	33.0	34.0	35.0	37.0	38.0	40.0	41.0	43.0	44.0	46.0	47.0	49.0	50.0	52.0	53.0	55.0	56.0	58.0	59.0	61.0	62.0	64.0	65.0	67.0
25	33.0	34.0	36.0	37.0	39.0	40.0	42.0	43.0	45.0	46.0	48.0	49.0	51.0	52.0	54.0	55.0	57.0	58.0	60.0	62.0	63.0	65.0	66.0	68.0	70.0
26	35.0	36.0	38.0	39.0	41.0	42.0	44.0	45.0	47.0	48.0	50.0	51.0	53.0	55.0	56.0	58.0	59.0	61.0	63.0	64.0	66.0	68.0	69.0	71.0	73.0
27	37.0	38.0	40.0	41.0	43.0	44.0	46.0	48.0	49.0	51.0	52.0	54.0	55.0	57.0	59.0	60.0	62.0	64.0	65.0	67.0	69.0	70.0	72.0	74.0	76.0
28	39.0	40.0	42.0	44.0	45.0	47.0	48.0	50.0	51.0	53.0	55.0	56.0	58.0	60.0	61.0	63.0	65.0	66.0	68.0	70.0	72.0	73.0	75.0	77.0	79.0
29	41.0	43.0	44.0	46.0	47.0	49.0	51.0	52.0	54.0	56.0	57.0	59.0	61.0	62.0	64.0	66.0	67.0	69.0	71.0	73.0	75.0	76.0	78.0	80.0	81.9
30	43.0	45.0	46.0	48.0	50.0	51.0	53.0	55.0	56.0	58.0	60.0	62.0	63.0	65.0	67.0	69.0	70.0	72.0	74.0	76.0	78.0	80.0	81.9	82.9	84.9
31	46.0	47.0	49.0	51.0	52.0	54.0	56.0	57.0	59.0	61.0	63.0	64.0	66.0	68.0	70.0	72.0	74.0	75.0	77.0	79.0	80.9	82.9	84.9	86.9	88.9
32	48.0	50.0	51.0	53.0	55.0	57.0	58.0	60.0	62.0	64.0	66.0	67.0	69.0	71.0	73.0	75.0	77.0	79.0	80.9	82.9	84.9	86.9	88.9	90.9	92.9
33	50.0	52.0	54.0	56.0	58.0	59.0	61.0	63.0	65.0	67.0	69.0	71.0	73.0	74.0	76.0	78.0	80.0	81.9	83.9	86.9	88.9	90.9	92.9	94.9	96.9
34	53.0	55.0	57.0	59.0	60.0	62.0	64.0	66.0	68.0	70.0	72.0	74.0	76.0	78.0	80.0	81.9	83.9	85.9	87.9	90.9	92.9	94.9	96.9	98.9	101.9
35	56.0	58.0	60.0	62.0	63.0	65.0	67.0	69.0	71.0	73.0	75.0	77.0	80.0	81.9	83.9	85.9	87.9	89.9	92.9	94.9	96.9	98.9	101.9	103.9	105.9
36	59.0	61.0	63.0	65.0	67.0	69.0	71.0	73.0	75.0	77.0	79.0	80.9	82.9	85.9	87.9	89.9	91.9	94.9	96.9	98.9	101.9	103.9	106.9	108.9	111.9
37	62.0	64.0	66.0	68.0	70.0	72.0	74.0	76.0	79.0	80.9	82.9	84.9	87.9	89.9	91.9	94.9	96.9	98.9	101.9	103.9	106.9	108.9	111.9	114.9	116.9
38	65.0	67.0	69.0	72.0	74.0	76.0	78.0	80.0	82.9	84.9	86.9	89.9	91.9	93.9	96.9	98.9	101.9	104.9	106.9	109.9	111.9	114.9	117.9	120.9	123.9
39	69.0	71.0	73.0	75.0	78.0	80.0	81.9	84.9	86.9	88.9	91.9	93.9	96.9	98.9	101.9	104.9	106.9	109.9	112.9	115.9	118.9	120.9	123.9	126.9	129.9
40	72.0	75.0	77.0	79.0	81.9	83.9	86.9	88.9	91.9	93.9	96.9	98.9	101.9	104.9	107.9	109.9	112.9	115.9	118.9	121.9	124.9	127.9	131.9	134.9	137.9
41	76.0	79.0	80.9	83.9	85.9	88.9	90.9	93.9	96.9	99.9	101.9	104.9	107.9	110.9	113.9	116.9	119.9	122.9	125.9	129.9	132.9	135.9	139.9	142.9	146.9
42	80.9	82.9	85.9	88.9	90.9	93.9	96.9	99.9	101.9	104.9	107.9	110.9	113.9	117.9	120.9	123.9	126.9	130.9	133.9	137.9	141.9	144.9	148.9	152.9	156.9
43	85.9	87.9	90.9	93.9	96.9	99.9	102.9	105.9	108.9	111.9	114.9	117.9	121.9	124.9	128.9	131.9	135.9	139.9	142.9	146.9	150.9	155.9	159.9	163.9	168.9
44	90.9	93.9	96.9	99.9	102.9	105.9	108.9	111.9	115.9	118.9	122.9	125.9	129.9	133.9	137.9	140.9	145.9	149.9	153.9	158.9	162.9	167.9	172.9	177.9	182.9
45	96.9	99.9	102.9	105.9	108.9	112.9	115.9	119.9	123.9	126.9	130.9	134.9	138.9	143.9	147.9	151.9	156.9	161.9	166.9	171.9	176.9	182.9	187.9	193.9	199.9
46	102.9	105.9	109.9	112.9	116.9	120.9	124.9	128.9	132.9	136.9	140.9	145.9	150.9	155.9	160.9	165.9	170.9	176.9	181.9	187.9	194.9	200.9	207.9	214.9	221.9
47	109.9	113.9	117.9	121.9	125.9	129.9	134.9	138.9	143.9	148.9	153.9	158.9	163.9	169.9	175.9	181.9	188.9	195.9	202.9	209.9	217.9	225.9	234.8	243.8	252.8
48	118.9	122.9	126.9	131.9	136.9	140.9	146.9	151.9	156.9	162.9	168.9	175.9	181.9	188.9	196.9	203.9	212.9	220.9	230.8	239.8	250.8	261.8	272.8	285.8	298.8
49	128.9	133.9	138.9	143.9	149.9	154.9	161.9	167.9	174.9	181.9	189.9	197.9	206.9	215.9	225.9	235.8	247.8	259.8	273.8	287.8	302.8	319.8	337.8	356.8	376.7

(MPNs with no shading have occurrence probability >0.01; lightly-shaded MPNs have occurrence probability between 10^{-2} and 10^{-6}; darker-shaded MPNs have occurrence probability less than 10^{-6})

Table 9 (ctd.): **Calculated MPN values for the 49x48 well IDEXX Quanti-Tray/2000™ (output from FORTRAN program Q4948).**

```
MPNs for 49 large well x 48 small well QuantiTray.
Large well volume = 1.86 mL, small well volume = .186 mL.
# large |
 wells  |
positive|                # small wells positive
        |   25     26     27     28     29     30     31     32     33     34     35     36     37     38     39     40     41     42     43     44     45     46     47     48
        |
    0   |  25.0   26.0   27.0   28.0   29.0   30.0   31.0   32.0   34.0   35.0   36.0   37.0   38.0   39.0   40.0   41.0   42.0   43.0   44.0   45.0   46.0   48.0   49.0   50.0
    1   |  26.0   27.0   28.0   30.0   31.0   32.0   33.0   34.0   35.0   36.0   37.0   38.0   39.0   40.0   41.0   43.0   44.0   45.0   46.0   47.0   48.0   49.0   50.0   51.0
    2   |  28.0   29.0   30.0   31.0   32.0   33.0   34.0   35.0   36.0   37.0   39.0   40.0   41.0   42.0   43.0   44.0   45.0   46.0   47.0   49.0   50.0   51.0   52.0   53.0
    3   |  29.0   30.0   31.0   32.0   33.0   35.0   36.0   37.0   38.0   39.0   40.0   41.0   42.0   43.0   45.0   46.0   47.0   48.0   49.0   50.0   51.0   52.0   54.0   55.0
    4   |  30.0   32.0   33.0   34.0   35.0   36.0   37.0   38.0   39.0   40.0   42.0   43.0   44.0   45.0   46.0   47.0   48.0   50.0   51.0   52.0   53.0   54.0   55.0   57.0
    5   |  32.0   33.0   34.0   35.0   36.0   37.0   39.0   40.0   41.0   42.0   43.0   44.0   45.0   47.0   48.0   49.0   50.0   51.0   52.0   54.0   55.0   56.0   57.0   58.0
    6   |  33.0   34.0   36.0   37.0   38.0   39.0   40.0   41.0   42.0   44.0   45.0   46.0   47.0   48.0   49.0   51.0   52.0   53.0   54.0   55.0   57.0   58.0   59.0   60.0
    7   |  35.0   36.0   37.0   38.0   39.0   41.0   42.0   43.0   44.0   45.0   46.0   48.0   49.0   50.0   51.0   52.0   54.0   55.0   56.0   57.0   58.0   60.0   61.0   62.0
    8   |  36.0   38.0   39.0   40.0   41.0   42.0   43.0   45.0   46.0   47.0   48.0   49.0   51.0   52.0   53.0   54.0   55.0   57.0   58.0   59.0   60.0   62.0   63.0   64.0
    9   |  38.0   39.0   40.0   41.0   43.0   44.0   45.0   46.0   48.0   49.0   50.0   51.0   52.0   54.0   55.0   56.0   57.0   59.0   60.0   61.0   62.0   64.0   65.0   66.0
   10   |  40.0   41.0   42.0   43.0   44.0   46.0   47.0   48.0   49.0   51.0   52.0   53.0   54.0   56.0   57.0   58.0   59.0   61.0   62.0   63.0   64.0   66.0   67.0   68.0
   11   |  41.0   42.0   44.0   45.0   46.0   47.0   49.0   50.0   51.0   52.0   54.0   55.0   56.0   57.0   59.0   60.0   61.0   63.0   64.0   65.0   66.0   68.0   69.0   70.0
   12   |  43.0   44.0   45.0   47.0   48.0   49.0   50.0   52.0   53.0   54.0   56.0   57.0   58.0   59.0   61.0   62.0   63.0   65.0   66.0   67.0   69.0   70.0   71.0   73.0
   13   |  45.0   46.0   47.0   49.0   50.0   51.0   52.0   54.0   55.0   56.0   58.0   59.0   60.0   62.0   63.0   64.0   66.0   67.0   68.0   70.0   71.0   72.0   74.0   75.0
   14   |  47.0   48.0   49.0   50.0   52.0   53.0   54.0   56.0   57.0   58.0   60.0   61.0   62.0   64.0   65.0   66.0   68.0   69.0   70.0   72.0   73.0   75.0   76.0   77.0
   15   |  48.0   50.0   51.0   52.0   54.0   55.0   56.0   58.0   59.0   60.0   62.0   63.0   65.0   66.0   67.0   69.0   70.0   71.0   73.0   74.0   76.0   77.0   79.0   80.0
   16   |  50.0   52.0   53.0   54.0   56.0   57.0   58.0   60.0   61.0   63.0   64.0   65.0   67.0   68.0   70.0   71.0   72.0   74.0   75.0   77.0   78.0   80.0   80.9   82.9
   17   |  52.0   54.0   55.0   56.0   58.0   59.0   61.0   62.0   63.0   65.0   66.0   68.0   69.0   71.0   72.0   73.0   75.0   76.0   78.0   79.0   80.9   81.9   83.9   84.9
   18   |  54.0   56.0   57.0   59.0   60.0   61.0   63.0   64.0   66.0   67.0   69.0   70.0   72.0   73.0   75.0   76.0   78.0   79.0   80.9   81.9   83.9   84.9   86.9   87.9
   19   |  57.0   58.0   59.0   61.0   62.0   64.0   65.0   67.0   68.0   70.0   71.0   73.0   74.0   76.0   77.0   79.0   80.0   81.9   82.9   84.9   85.9   87.9   89.9   90.9
   20   |  59.0   60.0   62.0   63.0   65.0   66.0   68.0   69.0   71.0   72.0   74.0   75.0   77.0   78.0   80.0   80.9   82.9   84.9   85.9   87.9   88.9   90.9   92.9   93.9
   21   |  61.0   63.0   64.0   66.0   67.0   69.0   70.0   72.0   73.0   75.0   76.0   78.0   80.0   80.9   82.9   83.9   85.9   87.9   88.9   90.9   92.9   93.9   95.9   97.9
   22   |  64.0   65.0   67.0   68.0   70.0   71.0   73.0   74.0   76.0   78.0   79.0   80.9   81.9   83.9   85.9   86.9   88.9   90.9   91.9   93.9   95.9   96.9   98.9  100.9
   23   |  66.0   68.0   69.0   71.0   72.0   74.0   76.0   77.0   79.0   80.0   81.9   83.9   84.9   86.9   88.9   90.9   91.9   93.9   95.9   96.9   98.9  100.9  102.9  104.9
   24   |  69.0   70.0   72.0   74.0   75.0   77.0   78.0   80.0   81.9   83.9   84.9   86.9   88.9   89.9   91.9   93.9   95.9   96.9   98.9  100.9  102.9  104.9  105.9  107.9
   25   |  71.0   73.0   75.0   76.0   78.0   80.0   81.9   82.9   84.9   86.9   87.9   89.9   91.9   93.9   95.9   96.9   98.9  100.9  102.9  104.9  106.9  108.9  109.9  111.9
   26   |  74.0   76.0   78.0   79.0   80.9   82.9   84.9   85.9   87.9   89.9   91.9   93.9   95.9   96.9   98.9  100.9  102.9  104.9  106.9  108.9  110.9  112.9  114.9  116.9
   27   |  77.0   79.0   80.9   82.9   83.9   85.9   87.9   89.9   91.9   93.9   94.9   96.9   98.9  100.9  102.9  104.9  106.9  108.9  110.9  112.9  114.9  116.9  118.9  120.9
   28   |  80.0   81.9   83.9   85.9   87.9   89.9   91.9   92.9   94.9   96.9   98.9  100.9  102.9  104.9  106.9  108.9  110.9  112.9  114.9  116.9  119.9  121.9  123.9  125.9
   29   |  83.9   85.9   87.9   88.9   90.9   92.9   94.9   96.9   98.9  100.9  102.9  104.9  106.9  108.9  111.9  113.9  115.9  117.9  119.9  121.9  123.9  126.9  128.9  130.9
   30   |  86.9   88.9   90.9   92.9   94.9   96.9   98.9  100.9  102.9  105.9  107.9  109.9  111.9  113.9  115.9  117.9  120.9  122.9  124.9  126.9  129.9  131.9  133.9  136.9
   31   |  90.9   92.9   94.9   96.9   98.9  100.9  103.9  105.9  107.9  109.9  111.9  113.9  116.9  118.9  120.9  123.9  125.9  127.9  130.9  132.9  134.9  137.9  139.9  142.9
   32   |  94.9   96.9   98.9  100.9  103.9  105.9  107.9  109.9  112.9  114.9  116.9  118.9  121.9  123.9  126.9  128.9  131.9  133.9  135.9  138.9  141.9  143.9  146.9  148.9
   33   |  98.9  101.9  103.9  105.9  107.9  110.9  112.9  114.9  117.9  119.9  122.9  124.9  126.9  129.9  131.9  134.9  137.9  139.9  142.9  145.9  147.9  150.9  153.9  156.9
   34   | 103.9  105.9  108.9  110.9  112.9  115.9  117.9  120.9  122.9  125.9  127.9  130.9  133.9  135.9  138.9  141.9  143.9  146.9  149.9  152.9  155.9  158.9  160.9  163.9
   35   | 108.9  110.9  113.9  115.9  118.9  120.9  123.9  126.9  128.9  131.9  134.9  136.9  139.9  142.9  145.9  148.9  151.9  154.9  157.9  160.9  163.9  166.9  169.9  172.9
   36   | 113.9  116.9  118.9  121.9  124.9  126.9  129.9  132.9  135.9  138.9  141.9  143.9  146.9  149.9  153.9  156.9  159.9  162.9  165.9  168.9  172.9  175.9  179.9  182.9
   37   | 119.9  122.9  125.9  127.9  130.9  133.9  136.9  139.9  142.9  145.9  148.9  151.9  155.9  158.9  161.9  165.9  168.9  171.9  175.9  178.9  182.9  186.9  189.9  193.9
   38   | 125.9  128.9  131.9  134.9  137.9  141.9  144.9  147.9  150.9  153.9  157.9  160.9  164.9  167.9  171.9  175.9  178.9  182.9  186.9  190.9  194.9  198.9  202.9  206.9
   39   | 133.9  136.9  139.9  142.9  145.9  149.9  152.9  156.9  159.9  163.9  167.9  170.9  174.9  178.9  182.9  186.9  190.9  194.9  198.9  203.9  207.9  212.9  216.9  221.9
   40   | 141.9  144.9  147.9  151.9  155.9  158.9  162.9  166.9  170.9  174.9  178.9  182.9  186.9  190.9  195.9  199.9  204.9  208.9  213.9  218.9  223.9  228.8  233.8  238.8
   41   | 150.9  153.9  157.9  161.9  165.9  169.9  173.9  178.9  182.9  186.9  191.9  195.9  200.9  205.9  210.9  215.9  220.9  225.9  231.8  236.8  242.8  248.8  253.8  259.8
   42   | 160.9  164.9  168.9  173.9  177.9  182.9  186.9  191.9  196.9  201.9  206.9  212.9  217.9  223.9  228.8  234.8  240.8  246.8  252.8  259.8  265.8  272.8  279.8  286.8
   43   | 172.9  177.9  182.9  187.9  192.9  197.9  202.9  208.9  214.9  219.9  225.9  231.8  238.8  244.8  251.8  258.8  265.8  272.8  279.8  287.8  295.8  303.8  311.8  319.8
   44   | 187.9  192.9  198.9  204.9  210.9  216.9  222.9  229.8  235.8  242.8  249.8  257.8  264.8  272.8  280.8  288.8  297.8  305.8  314.8  324.8  333.8  343.8  353.8  363.8
   45   | 205.9  212.9  218.9  225.9  232.8  240.8  247.8  255.8  264.8  272.8  281.8  290.8  299.8  309.8  319.8  329.8  340.8  351.8  362.8  374.8  386.7  398.7  411.7  424.7
   46   | 229.8  237.8  245.8  254.8  263.8  272.8  282.8  292.8  303.8  313.8  325.8  336.8  349.8  361.8  374.8  388.7  402.7  416.7  431.7  446.7  462.7  478.7  495.7  513.7
   47   | 262.8  272.8  283.8  295.8  307.8  319.8  332.8  346.8  360.8  374.8  390.7  406.7  423.7  440.7  458.7  476.7  496.7  516.7  537.6  559.6  582.6  606.6  631.6  657.6
   48   | 312.8  326.8  342.8  358.8  375.7  393.7  412.7  432.7  452.7  474.7  497.7  520.7  545.6  571.6  599.6  628.6  659.6  691.5  726.5  763.5  803.5  846.4  893.4  944.4
   49   | 398.7  421.7  445.7  471.7  498.7  527.6  557.6  590.6  624.6  661.6  701.5  744.5  791.5  842.4  899.4  962.4 1034.3 1117.2 1215.2 1335.1 1490.0 1707.8 2079.6>2079.6
```

(MPNs with no shading have occurrence probability >0.01; lightly-shaded MPNs have an occurrence probability between 10^{-2} and 10^{-6}; darker-shaded MPNs have an occurrence probability < 10^{-6})

Table 10: Calculated MPN values for the 49x48 well IDEXX Quanti-Tray/2000™ (output from FORTRAN program Q4948).

MPNs for 49 large well x 48 small well QuantiTray.
Large well volume = 1.88 mL, small well volume = .16 mL.

# large wells positive	# small wells positive																								
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0
1	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.1
2	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.1	27.1
3	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	23.0	24.0	25.1	26.1	27.1	28.1
4	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	21.0	22.0	23.0	24.0	25.1	26.1	27.1	28.1	29.1
5	5.0	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0	16.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.1	26.1	27.1	28.1	30.1	31.1
6	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	24.0	25.1	27.1	28.1	29.1	30.1	31.1	32.1
7	7.0	8.0	9.0	10.0	11.0	12.0	13.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	22.0	23.0	25.1	26.1	27.1	28.1	29.1	30.1	31.1	32.1	34.1
8	8.0	9.0	10.0	11.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0	21.0	23.0	24.0	25.1	26.1	27.1	28.1	29.1	30.1	32.1	33.1	34.1	35.1
9	9.0	10.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	21.0	22.0	23.0	24.0	25.1	26.1	27.1	29.1	30.1	31.1	32.1	33.1	34.1	35.1	37.1
10	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	20.0	21.0	22.0	23.0	24.0	25.1	26.1	28.1	29.1	30.1	31.1	32.1	34.1	35.1	36.1	37.1	38.1
11	12.0	13.0	14.0	15.0	16.0	17.0	19.0	20.0	21.0	22.0	23.0	24.0	26.1	27.1	28.1	29.1	30.1	32.1	33.1	34.1	35.1	36.1	37.1	39.1	40.1
12	13.0	14.0	15.0	17.0	18.0	19.0	20.0	21.0	22.0	24.0	25.1	26.1	27.1	28.1	29.1	31.1	32.1	33.1	34.1	35.1	37.1	38.1	39.1	40.1	42.1
13	14.0	16.0	17.0	18.0	19.0	20.0	21.0	23.0	24.0	25.1	26.1	27.1	29.1	30.1	31.1	32.1	33.1	35.1	36.1	37.1	38.1	40.1	41.1	42.1	43.1
14	16.0	17.0	18.0	19.0	20.0	22.0	23.0	24.0	25.1	26.1	28.1	29.1	30.1	31.1	33.1	34.1	35.1	36.1	38.1	39.1	40.1	41.1	43.1	44.1	45.1
15	17.0	18.0	19.0	21.0	22.0	23.0	24.0	26.1	27.1	28.1	29.1	31.1	32.1	33.1	34.1	36.1	37.1	38.1	39.1	41.1	42.1	43.1	44.1	46.1	47.1
16	18.0	20.0	21.0	22.0	23.0	25.1	26.1	27.1	28.1	30.1	31.1	32.1	33.1	35.1	36.1	37.1	39.1	40.1	41.1	42.1	44.1	45.1	46.1	48.1	49.1
17	20.0	21.0	22.0	24.0	25.1	26.1	27.1	29.1	30.1	31.1	33.1	34.1	35.1	36.1	38.1	39.1	40.1	42.1	43.1	44.1	46.1	47.1	48.1	50.1	51.1
18	21.0	23.0	24.0	25.1	26.1	28.1	29.1	30.1	32.1	33.1	34.1	36.1	37.1	38.1	39.1	41.1	42.1	43.1	45.1	46.1	48.1	49.1	50.1	52.1	53.1
19	23.0	24.0	26.1	27.1	28.1	29.1	31.1	32.1	33.1	35.1	36.1	37.1	39.1	40.1	41.1	43.1	44.1	45.1	47.1	48.1	50.1	51.1	52.1	54.1	55.1
20	25.1	26.1	27.1	28.1	30.1	31.1	32.1	34.1	35.1	36.1	38.1	39.1	41.1	42.1	43.1	45.1	46.1	47.1	49.1	50.1	52.1	53.1	54.1	56.1	57.1
21	26.1	27.1	29.1	30.1	31.1	33.1	34.1	36.1	37.1	38.1	40.1	41.1	42.1	44.1	45.1	47.1	48.1	50.1	51.1	52.1	54.1	55.1	57.1	58.1	60.1
22	28.1	29.1	31.1	32.1	33.1	35.1	36.1	37.1	39.1	40.1	42.1	43.1	44.1	46.1	47.1	49.1	50.1	52.1	53.1	55.1	56.1	58.1	59.1	61.1	62.1
23	30.1	31.1	32.1	34.1	35.1	36.1	38.1	39.1	41.1	42.1	44.1	45.1	47.1	48.1	49.1	51.1	52.1	54.1	55.1	57.1	58.1	60.1	61.1	63.1	65.1
24	31.1	33.1	34.1	36.1	37.1	38.1	40.1	41.1	43.1	44.1	46.1	47.1	49.1	50.1	52.1	53.1	55.1	56.1	58.1	59.1	61.1	62.1	64.1	66.1	67.1
25	33.1	35.1	36.1	38.1	39.1	40.1	42.1	43.1	45.1	46.1	48.1	49.1	51.1	52.1	54.1	56.1	57.1	59.1	60.1	62.1	63.1	65.1	67.1	68.1	70.1
26	35.1	37.1	38.1	40.1	41.1	43.1	44.1	46.1	47.1	49.1	50.1	52.1	53.1	55.1	56.1	58.1	60.1	61.1	63.1	64.1	66.1	68.1	69.1	71.1	73.1
27	37.1	39.1	40.1	42.1	43.1	45.1	46.1	48.1	49.1	51.1	53.1	54.1	56.1	57.1	59.1	61.1	62.1	64.1	66.1	67.1	69.1	71.1	72.1	74.1	76.2
28	39.1	41.1	42.1	44.1	45.1	47.1	49.1	50.1	52.1	53.1	55.1	57.1	58.1	60.1	62.1	63.1	65.1	67.1	68.1	70.1	72.1	74.1	75.2	77.2	79.2
29	41.1	43.1	44.1	46.1	48.1	49.1	51.1	53.1	54.1	56.1	58.1	59.1	61.1	63.1	64.1	66.1	68.1	70.1	71.1	73.1	75.2	77.2	79.2	80.2	82.2
30	44.1	45.1	47.1	48.1	50.1	52.1	53.1	55.1	57.1	59.1	60.1	62.1	64.1	65.1	67.1	69.1	71.1	73.1	74.1	76.2	78.2	80.2	82.2	84.2	86.2
31	46.1	48.1	49.1	51.1	53.1	54.1	56.1	58.1	60.1	61.1	63.1	65.1	67.1	68.1	70.1	72.1	74.1	76.2	78.2	80.2	82.2	84.2	86.2	88.2	90.2
32	48.1	50.1	52.1	54.1	55.1	57.1	59.1	61.1	62.1	64.1	66.1	68.1	70.1	72.1	74.1	75.2	77.2	79.2	81.2	83.2	85.2	87.2	89.2	91.2	94.2
33	51.1	53.1	54.1	56.1	58.1	60.1	62.1	64.1	65.1	67.1	69.1	71.1	73.1	75.2	77.2	79.2	81.2	83.2	85.2	87.2	89.2	91.2	93.2	96.2	98.2
34	54.1	55.1	57.1	59.1	61.1	63.1	65.1	67.1	69.1	71.1	73.1	75.2	77.2	79.2	81.2	83.2	85.2	87.2	89.2	91.2	93.2	96.2	98.2	100.2	102.2
35	56.1	58.1	60.1	62.1	64.1	66.1	68.1	70.1	72.1	74.1	76.2	78.2	80.2	82.2	85.2	87.2	89.2	91.2	93.2	96.2	98.2	100.2	103.2	105.2	107.2
36	59.1	61.1	63.1	65.1	67.1	69.1	71.1	74.1	76.2	78.2	80.2	82.2	84.2	87.2	89.2	91.2	93.2	96.2	98.2	100.2	103.2	105.2	108.2	110.2	113.2
37	63.1	65.1	67.1	69.1	71.1	73.1	75.2	77.2	80.2	82.2	84.2	86.2	89.2	91.2	93.2	96.2	98.2	101.2	103.2	106.2	108.2	111.2	113.2	116.2	119.2
38	66.1	68.1	70.1	72.1	75.2	77.2	79.2	81.2	84.2	86.2	88.2	91.2	93.2	96.2	98.2	101.2	103.2	106.2	108.2	111.2	114.2	117.2	119.2	122.2	125.3
39	70.1	72.1	74.1	76.2	79.2	81.2	83.2	86.2	88.2	91.2	93.2	96.2	98.2	101.2	104.2	106.2	109.2	112.2	114.2	117.2	120.2	123.2	126.3	129.3	132.3
40	74.1	76.2	78.2	81.2	83.2	86.2	88.2	91.2	93.2	96.2	98.2	101.2	104.2	107.2	109.2	112.2	115.2	118.2	121.2	124.2	127.3	131.3	134.3	137.3	140.3
41	78.2	80.2	83.2	85.2	88.2	90.2	93.2	96.2	98.2	101.2	104.2	107.2	110.2	113.2	116.2	119.2	122.2	125.3	129.3	132.3	135.3	139.3	142.3	146.3	150.3
42	82.2	85.2	88.2	90.2	93.2	96.2	99.2	101.2	104.2	107.2	110.2	114.2	117.2	120.2	123.2	127.3	130.3	134.3	137.3	141.3	144.3	148.3	152.3	156.3	160.3
43	87.2	90.2	93.2	96.2	99.2	102.2	105.2	108.2	111.2	114.2	118.2	121.2	124.2	128.3	132.3	135.3	139.3	143.3	147.3	151.3	155.3	159.3	164.3	168.3	173.3
44	93.2	96.2	99.2	102.2	105.2	108.2	112.2	115.2	119.2	122.2	126.3	129.3	133.3	137.3	141.3	145.3	150.3	154.3	158.3	163.3	168.3	173.3	178.4	183.4	188.4
45	99.2	102.2	106.2	109.2	112.2	116.2	120.2	123.2	127.3	131.3	135.3	139.3	144.3	148.3	153.3	157.3	162.3	167.3	173.3	178.4	183.4	189.4	195.4	201.4	208.4
46	106.2	109.2	113.2	117.2	121.2	125.3	129.3	133.3	137.3	142.3	146.3	151.3	156.3	161.3	167.3	172.3	178.4	184.4	190.4	197.4	204.4	211.4	218.4	225.5	233.5
47	114.2	118.2	122.2	126.3	131.3	135.3	140.3	145.3	150.3	155.3	160.3	166.3	172.3	178.4	185.4	192.4	199.4	206.4	214.4	222.4	231.5	240.5	249.5	259.5	270.5
48	123.2	128.3	133.3	138.3	143.3	148.3	153.3	159.3	165.3	172.3	179.4	186.4	193.4	201.4	209.4	218.4	228.5	238.5	249.5	260.5	272.5	285.6	298.6	313.6	328.7
49	135.3	140.3	146.3	152.3	158.3	164.3	171.3	179.4	187.4	195.4	204.4	214.4	224.4	235.5	247.5	260.5	275.6	290.6	307.6	324.6	344.7	364.7	386.8	409.8	434.9

(MPNs with no shading have occurrence probability >0.01; lightly-shaded MPNs have occurrence probability between 10^{-2} and 10^{-6}; darker-shaded MPNs have occurrence probability less than 10^{-6})

Table 10 (ctd): Calculated MPN values for the 49x48 well IDEXX Quanti-Tray/2000™ (output from FORTRAN program Q4948).

```
MPNs for 49 large well x 48 small well QuantiTray.
Large well volume = 1.88 mL, small well volume = .16 mL.
```

# large wells positive	# small wells positive 25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
0	25.1	26.1	27.1	28.1	29.1	30.1	31.1	32.1	33.1	34.1	36.1	37.1	38.1	39.1	40.1	41.1	42.1	43.1	44.1	45.1	46.1	47.1	48.1	49.1
1	26.1	27.1	28.1	29.1	31.1	32.1	33.1	34.1	35.1	36.1	37.1	38.1	39.1	40.1	41.1	42.1	43.1	44.1	46.1	47.1	48.1	49.1	50.1	51.1
2	28.1	29.1	30.1	31.1	32.1	33.1	34.1	35.1	36.1	37.1	38.1	39.1	41.1	42.1	43.1	44.1	45.1	46.1	47.1	48.1	49.1	50.1	52.1	53.1
3	29.1	30.1	31.1	32.1	33.1	34.1	35.1	37.1	38.1	39.1	40.1	41.1	42.1	43.1	44.1	45.1	47.1	48.1	49.1	50.1	51.1	52.1	53.1	54.1
4	30.1	31.1	33.1	34.1	35.1	36.1	37.1	38.1	39.1	40.1	41.1	43.1	44.1	45.1	46.1	47.1	48.1	49.1	50.1	52.1	53.1	54.1	55.1	56.1
5	32.1	33.1	34.1	35.1	36.1	37.1	38.1	40.1	41.1	42.1	43.1	44.1	45.1	46.1	48.1	49.1	50.1	51.1	52.1	53.1	54.1	56.1	57.1	58.1
6	33.1	34.1	35.1	37.1	38.1	39.1	40.1	41.1	42.1	43.1	45.1	46.1	47.1	48.1	49.1	50.1	52.1	53.1	54.1	55.1	56.1	57.1	59.1	60.1
7	35.1	36.1	37.1	38.1	39.1	40.1	42.1	43.1	44.1	45.1	46.1	47.1	49.1	50.1	51.1	52.1	53.1	55.1	56.1	57.1	58.1	59.1	61.1	62.1
8	36.1	37.1	39.1	40.1	41.1	42.1	43.1	44.1	46.1	47.1	48.1	49.1	50.1	52.1	53.1	54.1	55.1	56.1	58.1	59.1	60.1	61.1	62.1	64.1
9	38.1	39.1	40.1	41.1	43.1	44.1	45.1	46.1	47.1	49.1	50.1	51.1	52.1	53.1	55.1	56.1	57.1	58.1	60.1	61.1	62.1	63.1	64.1	66.1
10	39.1	41.1	42.1	43.1	44.1	45.1	47.1	48.1	49.1	50.1	52.1	53.1	54.1	55.1	57.1	58.1	59.1	60.1	62.1	63.1	64.1	65.1	67.1	68.1
11	41.1	42.1	44.1	45.1	46.1	47.1	48.1	50.1	51.1	52.1	53.1	55.1	56.1	57.1	58.1	60.1	61.1	62.1	64.1	65.1	66.1	67.1	69.1	70.1
12	43.1	44.1	45.1	47.1	48.1	49.1	50.1	52.1	53.1	54.1	55.1	57.1	58.1	59.1	61.1	62.1	63.1	64.1	66.1	67.1	68.1	70.1	71.1	72.1
13	45.1	46.1	47.1	48.1	50.1	51.1	52.1	54.1	55.1	56.1	57.1	59.1	60.1	61.1	63.1	64.1	65.1	67.1	68.1	69.1	71.1	72.1	73.1	75.2
14	46.1	48.1	49.1	50.1	52.1	53.1	54.1	56.1	57.1	58.1	59.1	61.1	62.1	63.1	65.1	66.1	67.1	69.1	70.1	72.1	73.1	74.1	76.2	77.2
15	48.1	50.1	51.1	52.1	54.1	55.1	56.1	58.1	59.1	60.1	62.1	63.1	64.1	66.1	67.1	68.1	70.1	71.1	73.1	74.1	75.2	77.2	78.2	80.2
16	50.1	52.1	53.1	54.1	56.1	57.1	58.1	60.1	61.1	62.1	64.1	65.1	67.1	68.1	69.1	71.1	72.1	74.1	75.2	76.2	78.2	79.2	81.2	82.2
17	52.1	54.1	55.1	56.1	58.1	59.1	61.1	62.1	63.1	65.1	66.1	68.1	69.1	70.1	72.1	73.1	75.2	76.2	78.2	79.2	81.2	82.2	83.2	85.2
18	54.1	56.1	57.1	59.1	60.1	61.1	63.1	64.1	66.1	67.1	69.1	70.1	71.1	73.1	74.1	76.2	77.2	79.2	80.2	82.2	83.2	85.2	86.2	88.2
19	57.1	58.1	59.1	61.1	62.1	64.1	65.1	67.1	68.1	70.1	71.1	72.1	74.1	75.2	77.2	78.2	80.2	82.2	83.2	85.2	86.2	88.2	89.2	91.2
20	59.1	60.1	62.1	63.1	65.1	66.1	68.1	69.1	71.1	72.1	74.1	75.2	77.2	78.2	80.2	81.2	83.2	84.2	86.2	88.2	89.2	91.2	92.2	94.2
21	61.1	63.1	64.1	66.1	67.1	69.1	70.1	72.1	73.1	75.2	76.2	78.2	79.2	81.2	83.2	84.2	86.2	87.2	89.2	91.2	92.2	94.2	96.2	97.2
22	64.1	65.1	67.1	68.1	70.1	71.1	73.1	74.1	76.2	78.2	79.2	81.2	82.2	84.2	86.2	87.2	89.2	91.2	92.2	94.2	96.2	97.2	99.2	101.2
23	66.1	68.1	69.1	71.1	72.1	74.1	76.2	77.2	79.2	80.2	82.2	84.2	85.2	87.2	89.2	90.2	92.2	94.2	96.2	97.2	99.2	101.2	102.2	104.2
24	69.1	70.1	72.1	74.1	75.2	77.2	79.2	80.2	82.2	84.2	85.2	87.2	89.2	90.2	92.2	94.2	96.2	97.2	99.2	101.2	103.2	104.2	106.2	108.2
25	72.1	73.1	75.2	77.2	78.2	80.2	82.2	83.2	85.2	87.2	88.2	90.2	92.2	94.2	96.2	97.2	99.2	101.2	103.2	105.2	106.2	108.2	110.2	112.2
26	74.1	76.2	78.2	80.2	81.2	83.2	85.2	87.2	88.2	90.2	92.2	94.2	96.2	97.2	99.2	101.2	103.2	105.2	107.2	109.2	111.2	112.2	114.2	116.2
27	77.2	79.2	81.2	83.2	85.2	86.2	88.2	90.2	92.2	94.2	96.2	97.2	99.2	101.2	103.2	105.2	107.2	109.2	111.2	113.2	115.2	117.2	119.2	121.2
28	81.2	83.2	84.2	86.2	88.2	90.2	92.2	94.2	96.2	97.2	99.2	101.2	103.2	105.2	107.2	109.2	111.2	113.2	115.2	117.2	119.2	122.2	124.2	126.3
29	84.2	86.2	88.2	90.2	92.2	94.2	96.2	98.2	100.2	101.2	104.2	106.2	108.2	110.2	112.2	114.2	116.2	118.2	120.2	122.2	124.2	127.3	129.3	131.3
30	88.2	90.2	92.2	94.2	96.2	98.2	100.2	102.2	104.2	106.2	108.2	110.2	112.2	114.2	116.2	119.2	121.2	123.2	125.3	128.3	130.3	132.3	134.3	137.3
31	92.2	94.2	96.2	98.2	100.2	102.2	104.2	106.2	108.2	110.2	113.2	115.2	117.2	119.2	122.2	124.2	126.3	128.3	131.3	133.3	136.3	138.3	140.3	143.3
32	96.2	98.2	100.2	102.2	104.2	106.2	109.2	111.2	113.2	115.2	118.2	120.2	122.2	125.3	127.3	129.3	132.3	134.3	137.3	139.3	142.3	144.3	147.3	149.3
33	100.2	102.2	104.2	107.2	109.2	111.2	114.2	116.2	118.2	121.2	123.2	125.3	128.3	130.3	133.3	136.3	138.3	141.3	143.3	146.3	149.3	151.3	154.3	157.3
34	105.2	107.2	109.2	112.2	114.2	116.2	119.2	121.2	124.2	126.3	129.3	132.3	134.3	137.3	139.3	142.3	145.3	148.3	150.3	153.3	156.3	159.3	162.3	165.3
35	110.2	112.2	115.2	117.2	120.2	122.2	125.3	127.3	130.3	133.3	135.3	138.3	141.3	144.3	147.3	150.3	152.3	155.3	158.3	161.3	164.3	168.3	171.3	174.3
36	115.2	118.2	120.2	123.2	126.3	129.3	131.3	134.3	137.3	140.3	143.3	146.3	149.3	152.3	155.3	158.3	161.3	164.3	167.3	170.3	174.3	177.4	180.4	184.4
37	121.2	124.2	127.3	130.3	133.3	135.3	138.3	141.3	144.3	147.3	151.3	154.3	157.3	160.3	163.3	167.3	170.3	174.3	177.4	181.4	184.4	188.4	192.4	195.4
38	128.3	131.3	134.3	137.3	140.3	143.3	146.3	150.3	153.3	156.3	160.3	163.3	166.3	170.3	174.3	177.4	181.4	185.4	188.4	192.4	196.4	200.4	204.4	208.4
39	135.3	139.3	142.3	145.3	148.3	152.3	155.3	159.3	162.3	166.3	170.3	173.3	177.4	181.4	185.4	189.4	193.4	197.4	202.4	206.4	210.4	215.4	219.4	224.4
40	144.3	147.3	151.3	154.3	158.3	162.3	166.3	169.3	173.3	177.4	181.4	186.4	190.4	194.4	199.4	203.4	208.4	212.4	217.4	222.4	227.5	232.5	237.5	242.5
41	153.3	157.3	161.3	165.3	169.3	173.3	177.4	182.4	186.4	191.4	195.4	200.4	205.4	210.4	215.4	220.4	225.5	230.5	236.5	241.5	247.5	252.5	258.5	264.5
42	164.3	169.3	173.3	178.4	182.4	187.4	192.4	196.4	201.4	207.4	212.4	217.4	223.4	228.5	234.5	240.5	246.5	252.5	258.5	265.5	271.5	278.6	285.6	292.6
43	178.4	183.4	187.4	193.4	198.4	203.4	209.4	214.4	220.4	226.5	232.5	239.5	245.5	252.5	259.5	265.5	273.5	280.6	287.6	295.6	303.6	311.6	319.6	327.7
44	194.4	199.4	205.4	211.4	217.4	224.4	230.5	237.5	244.5	251.5	259.5	266.5	274.5	282.6	290.6	299.6	307.6	316.6	325.7	335.7	344.7	354.7	364.7	374.7
45	214.4	221.4	228.5	235.5	243.5	251.5	259.5	267.5	276.6	285.6	294.6	303.6	313.6	323.6	333.7	344.7	355.7	366.7	378.8	389.8	401.8	414.8	426.9	439.9
46	242.5	250.5	259.5	268.5	278.6	288.6	299.6	309.6	321.6	332.7	344.7	356.7	369.7	382.8	396.8	409.8	424.8	438.9	454.9	469.9	486.0	502.0	519.0	537.1
47	281.6	292.6	305.6	317.6	330.7	344.7	358.7	373.7	388.8	404.8	420.8	437.9	455.9	473.9	493.0	512.0	532.1	553.1	574.1	597.2	620.2	644.3	668.3	694.4
48	344.7	361.7	378.8	397.8	416.8	436.9	457.9	480.0	502.0	526.1	550.1	576.2	603.2	631.3	660.3	691.4	723.4	757.5	794.6	832.7	873.7	917.8	964.9	1016.0
49	460.9	488.0	517.0	547.1	579.2	612.2	648.3	686.4	726.5	769.5	815.6	865.7	919.8	980.0	1045.1	1119.2	1202.4	1298.6	1412.8	1552.1	1731.5	1985.0	2417.8	>2417.8

(MPNs with no shading have occurrence probability >0.01; lightly-shaded MPNs have an occurrence probability between 10^{-2} and 10^{-6}; darker-shaded MPNs have an occurrence probability < 10^{-6})

Table 11: **MPN values for the 51-Well IDEXX Quanti-Tray™ procedures.**

Number of wells giving a positive result	**Quanti-Tray™ Table**	**XactMPN**
0	0.0	0
1	1.0	1
2	2.0	2
3	3.1	3
4	4.2	4
5	5.3	5
6	6.4	6
7	7.5	7
8	8.7	8
9	9.9	9
10	11.1	11
11	12.4	12
12	13.7	13
13	15.0	14
14	16.4	16
15	17.8	17
16	19.2	19
17	20.7	20
18	22.2	22
19	23.8	23
20	25.4	25
21	27.1	26
22	28.8	28
23	30.6	30
24	32.4	32
25	34.4	34
26	36.4	36
27	38.4	38
28	40.6	40
29	42.9	42
30	45.3	45
31	47.8	47
32	50.4	50
33	53.1	52
34	56.0	55
35	59.1	58
36	62.4	62
37	65.9	65
38	69.7	69
39	73.8	73
40	78.2	77
41	83.1	82
42	88.5	88
43	94.5	94
44	101.3	100
45	109.1	108
46	118.4	117
47	129.8	129
48	144.5	143
49	165.2	164
50	200.5	199
51	>200.5	>199

Well volume = 100/51 ≈ 1.96 mL. Computed MPNs are integers, because total sample volume = 100 mL

***Table 12*:** **Confidence and credible 95% intervals (per 100 mL) for a 3x10 mL, 3x1 mL, 3x0.1 mL setup.**

	95% confidence intervals		95% credible intervals					
Pattern	**A**	**B**	**C**^{a}	**D**^{a}	**E**^{a}	**D**^{b}	**E**^{b}	**F**^{a}
0-1-0	0.085–13	0.1–10	<1.0–17	0.7–17	0.74–16.99	0.1–15	0.13–14.53	3–6
1-0-0	0.085–20	0.2–17	<1.0–21	0.9–21	0.87–20.59	0.1–18	0.14–17.53	3–6
1-0-1	0.87–21	1.2–17	2–27	2.2–27	2.24–27.11	1.1–24	1.06–23.89	6–12
1-1-0	0.88–23	1.3–20	2–28	2.2–28	2.28–27.62	1.1–24	1.08–24.33	6–12
2-0-0	1–36	1.5–35	2–38	2.9–38	2.87–37.73	1.3–33	1.26–32.82	6–12
1-2-0	2.7–36	4–35	4–35	4.1–35	4.14–34.68	2.6– 31	2.63–31.24	9–15
2-0-1	2.7–37	4–35	5–48	5.2– 48	5.21–47.92	3.1–43	3.10–42.60	9–21
2-1-0	2.8–44	4–38	5–50	5.3–50	5.35–49.50	3.2–44	3.17–43.97	9–21
2-2-0	3.5–47	5–40	8–62	8.5–63	8.46–62.59	5.8–56	5.77–56.43	12–30
3-0-0	3.5–120	5–94	<10–130	8.7–130	8.70–128.22	3.8–108	3.81–108.29	9–33
3-0-1	6.9–130	9–104	10–180	15–180	15.10–176.60	8.1–150	8.07–153.25	18–57
3-1-0	7.1–210	9–181	10–210	17–210	16.64–213.24	8.4–180	8.42–183.73	18–60
3-1-1	14–230	17–199	20–280	28^c–280	17.75–280.85	16–250	16.18–247.62	42–102
3-2-0	15–380	18–360	30–380	33–390	33.28–385.56	18–340	17.70–335.16	57–123
3-1-2	30–380	30–360	40–350	44–350	43.79–351.53	29–320	28.76–316.00	78–150
3-2-1	30–440	30–380	50–500	56–510	55.66–505.72	34–450	33.71–448.11	105–189
3-2-2	35–470	30–400	80–640	86–640	86.50–640.56	59–580	58.83–576.84	162–264
3-3-0	36–1300	40–990	<100–1400	91–1400	91.22–1395.58	40–1170	39.70–1174.56	183–291
3-3-1	71–2400	90–1980	100–2400	180–2400	178.29–2407.49	88–2070	87.52–2068.65	384–534
3-3-2	150–4800	200–4000	300–4800	380–4800	382.08–4788.17	190–4130	191.35–4133.64	981–1212

A: Woodward (1957); **B**: de Man (1983); **C**: de Man (1977); **D**: Beliaeff & Mary (1993) ; **E**: Using **Greenwood_Yule_MPN** (see Appendix C); **F**: Dalgety (1999).

[a] Central credible interval (with an area of 0.025 *in each tail* of the posterior distribution).

[b] Noncentral HPDR (shortest posterior interval with *total tail area* of 0.05 in the posterior distribution).

[c] This result may be a typographical error in Beliaeff & Mary (1993); arguably it should have been 18, not 28.

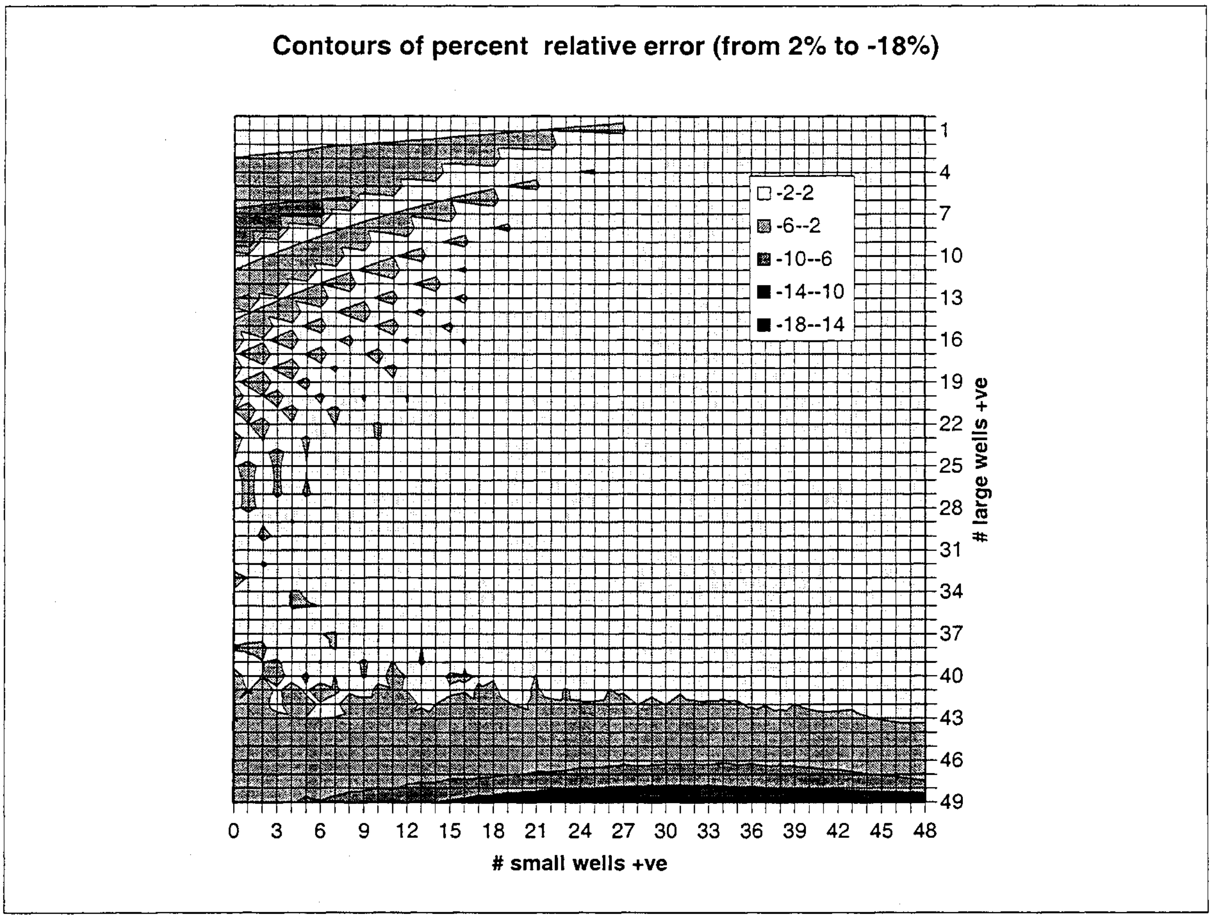

Figure 1: Relative errors between exact and IDEXX MPNs for Quanti-Tray/2000™ for large well volume = 1.86 mL and small cell volume = 0.186 mL (In the legend: '-2–2' means between -2 and 2; '-6—2' means between -6 and -2).

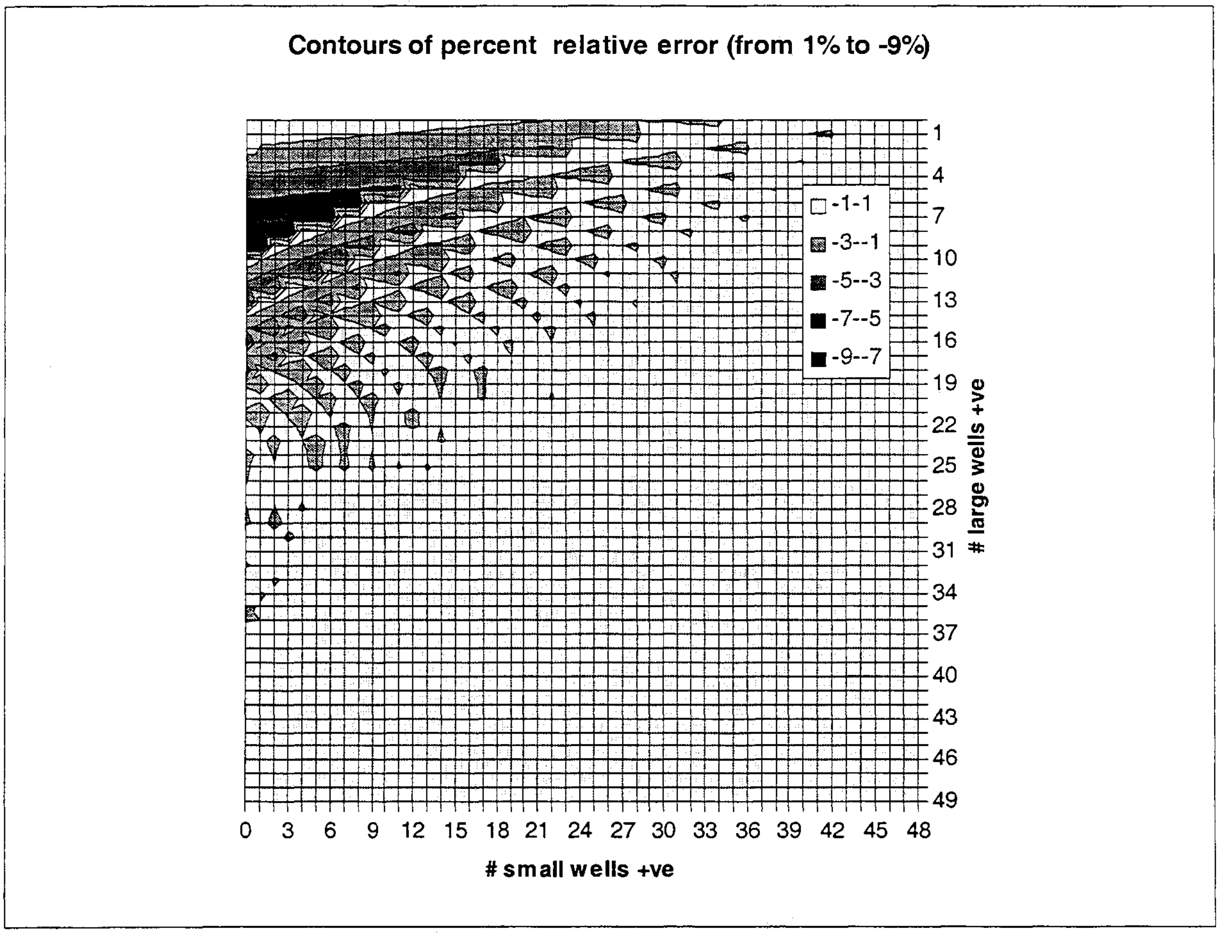

Figure 2: **Relative errors between exact and IDEXX MPNs for Quanti-Tray/2000™ for large well volume = 1.88 mL and small cell volume = 0.16 mL (In the legend: '-1–1' means between -1 and 1; '-5—3' means between -5 and -3).**

Appendix A: Computational details for exact MPNs

For multiple dilution series described by Equations 11–13 (in the main text) a considerable number of probabilities may have to be summed. Accordingly, to minimise effects of computer rounding errors all computations have been performed in double precision. Note that for very large n (typically for $n > 1500$) the factorial ratio term in those equations can become larger than a PC computer can cope with (i.e., $> 1.8\text{x}10^{308}$). But because that term multiplies other terms that are exceedingly small the overall term is itself less than unity (being a probability) and, using Equation 11 as an example, we can completely avoid any possibility of overflows by reformulating the equation to derive the following procedure:

$$\text{if } \zeta > 0 \text{ then } \Pr(i,j,k|n) = \sum_{n_I=0}^{n}\sum_{n_J=0}^{n-n_I}\exp(\xi); \text{ else} \Pr(i,j,k|n) = 0 \tag{A.1}$$

where

$$\zeta = \Pr(i|I,n_I)\Pr(j|J,n_J)\Pr(k|K,n_K) \tag{A.2}$$

with $n_K = n - (n_I + n_J)$, and

$$\xi = \ell n\left[\zeta \frac{n!}{n_I!n_J!n_K!} p_I^{n_I} p_J^{n_J} p_K^{n_K}\right] \tag{A.3}$$

which is evaluated as

$$\xi = \ell n(\zeta) + \ell n(n!) + \sum_{R=I,J,K} n_R \ell n(p_R) - \sum_{R=I,J,K} \ell n(n_R!) \tag{A.4}$$

in which the sums are first calculated separately, so that only one subtraction is needed. This device replaces multiplications and divisions of large and small numbers by sums and a difference of much smaller numbers (i.e., their logarithms).

The $\ell\text{n}(n!)$ terms in Equation A.4 may, of course, be simply evaluated [as $\ell\text{n}(n) + \ell\text{n}(n-1) + \ldots + \ell\text{n}(2)$], but as n gets large this method of computation becomes very slow. Herein we use a much faster technique, by noting that factorials are a special case of the gamma function (i.e., $\Gamma(n+1) = \text{n}!$, Abramowitz & Stegun (1972)). Press et al. (1986) gave very efficient FORTRAN code to calculate the logarithm of this function (the FACTLN and GAMMLN functions), based on the remarkably accurate formula of Lanczos (1962).

Evaluating the probabilities $\Pr(r\,|\,R,n)$ given by Equation 1 for each set of tubes is straightforward using the recursion formula (Equation 8); all terms are additive and nothing is too large. But because it is recursive on n, computation time becomes exceedingly long for large n. Evaluation of the direct Equation 9 is much quicker. However, considerable care is needed in its evaluation. Once again, for large n it is a mixture of extremely small and large terms and so logarithms must be used. But this is not straightforward either; we have to evaluate an oscillating series, and so issues of 'subtractive cancellation' arise, whereby computer rounding errors can swamp a series of subtractions with error, unbeknown to the novice (Dorn & McCracken 1972). These problems have been resolved first by grouping all terms in Equation 9 under the summation sign, i.e.,

$$\Pr(r|R,n)=\sum_{t=0}^{r-1}(-1)^t\exp(\eta)\quad \text{for } j\ (\le n) = 1, 2, \ldots, R \tag{A.5}$$

where

$$\eta=\ell n(R!)+n\ell n\left(\frac{r-t}{R}\right)-\{\ell n[(R-r)!]+\ell n(t!)+\ell n[(r-t)!]\} \tag{A.6}$$

and second, by summing the even and odd terms separately and then only subtracting them once (when all terms have been evaluated). This algorithm may be summarised as:

$$\text{for } r \text{ even: } \Pr_{even}(r|R,n)=\sum_{t=0(2)}^{r-2}\exp(\eta);\quad \Pr_{odd}(r|R,n)=\sum_{t=1(2)}^{r-1}\exp(\eta) \tag{A.7a}$$

$$\text{for } r \text{ odd: } \Pr_{even}(r|R,n)=\sum_{t=0(2)}^{r-1}\exp(\eta);\quad \Pr_{odd}(r|R,n)=\sum_{t=1(2)}^{r-2}\exp(\eta) \tag{A.7b}$$

with the subscripts $_{even}$ and $_{odd}$ referring to values of the summation's counting variable (i.e., t). The required probability is then simply given by

$$\Pr(r|R,n)=\Pr_{even}(r|R,n)-\Pr_{odd}(r|R,n) \tag{A.8}$$

Tiny oscillations can arise with this procedure (because of repeated application of the Lanczos formula, which has maximum errors on the order of 10^{-10} (Press et al. 1986). For example, comparing both procedures with the very accurate evaluation of lower values of Stirling numbers available in Mathematica™ (Wolfram 1992) for two test cases, we obtain the following results demonstrating this property.

Method	**Pr(r | R, n)**	
	$r = 6, R = 8, n = 10$	**$r = 2, R = 6, n = 12$**
Direct algorithm	0.4285874963066389	0.00002821136454107241
Recursive algorithm	0.4285874962806702	0.00002821136453764386
Mathematica™	0.4285874962806702	0.00002821136453764388

Comparing the recursive algorithm's results with those of Mathematica™, we see that there is full agreement down to the 15th or 16th significant digit, whereas the direct algorithm's results start to diverge from the other two at about the 10th digit.

Computer programs

The **XactMPN** program computes MPNs for up to four sets of tubes (or wells) using the procedures discussed above. Source code (in FORTRAN90) and executable file are available from the author (g.mcbride@niwa.co.nz); the source listing is given in Appendix B. The source code can be compiled on any machine supporting a FORTRAN compiler. The only non-standard FORTRAN feature is the use of the backslash (\) formatting character (allowing input data to be typed on the same line as its prompt). Most compilers will handle that.

The code is fully commented, explaining the detail of the logic used. A reasonable amount of error trapping has been built in to minimise the occurrence of crashes when presented with invalid input.

To avoid errors that may arise with the direct algorithm,[13] but making full use of its vastly shorter calculation time at large n values (see below), a two-step strategy is adopted:

a) use the direct algorithm first;
b) if the maximum probability calculated is less than a conservative threshold (taken as 10^{-6}), repeat all the calculations using only the recursive algorithm.

Every session with the program generates an output file (as well as output shown on-screen) for subsequent viewing. Output always echoes all the input data. On-screen instructions call for the following input data:

- output file number;
- number of sets of tubes or wells;
- mL of sample filtered or placed in each tube or well;
- number of positive tubes or wells in each set;
- lower and upper values of possible MPNs.

Making the lower possible MPN value close to its expected value can greatly reduce calculation time when that expected number is high (i.e., when nearly all the tubes are positive).

The 49x48 IDEXX Quanti-Tray/2000™ case was calculated using a stripped down version of **XactMPN** (called **Q4948**). Rather than starting each MPN search from $n = 0$, the initial guess at the MPN for each of the 2351 (i, j) combinations[14] is taken as that predicted using the formula presented by Thomas (1942).[15] If the next MPN has a lower occurrence probability the calculations will have commenced after the mode and so the initial MPN guess is reset to 99% of the first value. This procedure is repeated until the MPN next to the initial guess has a higher occurrence probability. Because some of the MPNs are highly improbable (i.e., occurrence probability less than 10^{-6}), one must use the recursive algorithm to calculate some parts of the table, and so the code can take a few hours to complete all the calculations.

Calculation time

The total number of calculation loops (N) required to compute the 49x48 IDEXX table, using either algorithm, is approximately

$$N = \sum_{i=0}^{I} \sum_{j=0}^{J} \left(\Delta n_{i,j} \nu_{i,j} \right) \qquad \text{(A.9)}$$

where

---

[13] Small truncation errors in the direct algorithm can give rise to noisy results resulting in false multi-modal distributions at very low occurrence probabilities, confounding the search for the true (single) mode (for a highly improbable MPN).

[14] That is, 49x48 – 1 (the $i = 49, j = 48$ case cannot be computed, because all wells are then positive).

[15] That is, $100(i + j)/\sqrt{(V_{neg} V_{total})}$, where V_{neg} is mL of sample volume in the negative wells, and V_{total} is total sample mL, see Equation 14.

$\Delta n_{i,j}$ is the difference between the initial guessed MPN and the final MPN (which we may take as n because all of the 100 mL is measured); and

$\nu_{i,j}$ is the number of loops required for either the direct or the recursive algorithm.

For the direct algorithm

$$\nu_{i,j} \approx (i-1) + (j-1) \tag{A.10}$$

while for the recursive algorithm

$$\nu_{i,j} \approx n(i+j) \tag{A.11}$$

In other words the direct algorithm is faster by a factor of about n.

This factor is particularly severe for large MPN values, because the number of loops is enormous using the recursive algorithm. For example, take the 49x47 positive well combination with MPN ≈ 2417 (as in Table 9). The initial guessed n, using the Thomas formula (Equation 14), is Int{100(49+47)/√[(49x1.88 + 48x0.16)(1x0.16)]} = 2402, so that $\Delta n_{49,47} = 15$. Therefore the recursive algorithm would call for 15x2417x(49+47) ≈ $3.5\text{x}10^6$ loops. In contrast the direct algorithm would call for only 15x(48+46) ≈ $1.4\text{x}10^3$ loops.

This particular MPN was calculated using the direct algorithm. This is because its occurrence probability is less than 10^{-6} (as shown on Table 9), and so the above rules do not require use of the slower recursive algorithm. In contrast, slightly lower MPNs (e.g., for positive well combinations 48x40 – 48x49) also have lower occurrence probabilities — less than 10^{-6} — and so their calculation used the much slower recursive algorithm. There is therefore a case for always using the direct algorithm, and simply not reporting a result if the calculated occurrence probability is less than 10^{-6}.

Similar arguments apply to the calculation of a four series, five replicate setup with most wells positive (e.g., some entries in the table presented by Haas & Heller (1988)).

Appendix B: Fortran computer code for XactMPN

```fortran
PROGRAM XactMPN
! Calculates exact MPNs for up to four sets of tubes (or wells), using
! occupancy theory, by repetitive calculation of occurrence probabilities for
! a range of candidate MPNs. The MPN is selected as the mode of those
! probabilities.

! Procedures are based on ideas and equations (for three sets of tubes) in
! Tillett & Coleman (1985, J. Appl. Bact. 59: 381-388). That is:
!
!  Pr(i,j,k|n) = occurrence probability, of getting i, j & k positive tubes in
!                a set of I, J & K tubes if a total of n bacteria were present
!
!      n  n-nI  (                                   n!       nI   nJ   nK  )
!  =  Sum  Sum  | Pr(i|I,nI)Pr(j|J,nJ)Pr(k|K,nK) --------- pI   pJ   pK    |...(1)
!    nI=0 nJ=0  (                                nI!nJ!nK!                 )
!
! where:
!  n           = total number of bacteria in the set of tubes (each n is a
!                candidate MPN);
!  nI, nJ, nK = numbers of bacteria in sets of tubes I, J & K;
!  nK          = n - (nI + nJ);
!  pI, pJ, pK = proportion of total sample volume (or filtrate) in sets I, J
!                & K;
!  Pr(r|R,nR) = probability of getting r positive tubes in a set of R tubes
!                containing equal volumes of sample or filtrate, where nR
!                bacteria are randomly distributed among them--one bacterium
!                being sufficient to guarantee a positive tube. These
!                probabilities are calculated in the FUNCTION "PrrRn", using an
!                old result in occupancy theory (David & Barton, 1962,
!                "Combinatorial Chance", Charles Griffin, London, p. 242). That
!                function uses a fast direct algorithm or a slower, but more
!                accurate, recursive algorithm. This program ("XactMPN") calls
!                PrrRn first with the direct algorithm, but then repeats the
!                call with the recursive if results are not accurate enough.

! If there are four sets of tubes the equation is
!
!                    n   n-nI  n-(nI+nJ)  (   )
!  Pr(i,j,k,l|n) =  Sum   Sum    Sum      | Z |, where
!                  nI=0  nJ=0    nK=0     (   )
!
!                                                   n!        nI  nJ  nK  nL
!  Z = Pr(i|I,nI)Pr(j|J,nJ)Pr(k|K,nK)Pr(L|L,nL)------------pI   pJ  pK  pL ...(2)
!                                              nI!nJ!nK!nL!
!
! where nL = n - (nI + nJ + nK).

! If there are only two sets of tubes the equation is
!
!                n  (                         n!      nI  nJ )
!  Pr(i,j|n) =  Sum | Pr(i|I,nI) Pr(j|J,nJ) ------ pI  pJ    |................(3)
!              nI=0 (                       nI!nJ!           )
!
! where nJ = n - nI.

! If there is only one set, the occurrence probability is simply given by
!
!    Pr(i|I,n).................................................................(4)
!
! which is calculated in the FUNCTION PrrRn.

! The MPN is the value of n at mode of the (smooth, unimodal) distribution of
! these probabilities over a range of n, divided by the sample volume.

! For efficiency in the code, I is subscripted so that I(1) = I, I(2) = J,
! I(3) = K, I(4) = L.  To avoid confusions between i and I etc. We use
! subscripted c as the counter for positive tubes, i.e., c(1) = i, c(2) = j,
! c(3) = k, c(4) = l.  Individual tube volumes ("Vtube") are also subscripted.
! Vtube is constant for each set of tubes. The integer counting dummy variable
! (used in DO loops) is m. Pr(i,j,k,l|n) [or Pr(i,k,k|n), etc.] is abbreviated
! to "OccurProb" (i.e., Occurrence probability).

! Uses double precision.  Evaluates ratios of large factorial numbers as
! exponentiated differences of factorials, to avoid numerical overflows.
! Log factorials are taken from the codes in Press et al. (1986, "Numerical
! Recipes", Cambridge: CUP), which use Lanzcos' formula for the required
! logarithm of the gamma function.  This is FAR faster than the more obvious
! direct evaluation of sums of logarithms.  The code is already slow enough
! for cases with high MPNs and lots of replicates!
```

```
! Programmed in MS FORTRAN by Graham McBride, NIWA, Hamilton, NZ, 10/98,
! revised 2/02. Fax: 64 7 856-0151; email: g.mcbride@niwa.co.nz

! Specifications
   IMPLICIT NONE
   DOUBLE PRECISION FACTLN,GAMMLN,MPNlow,MPNhigh,MPN_per_100mL,MPN_per_L
   DOUBLE PRECISION OldOccurProb,OccurProb,OccurProbmax,pI,pJ,pK,pL
   DOUBLE PRECISION PrrRn,TrustPr,Vtube(4),Vtotal,xi,xi1,xi2,zeta
   INTEGER          ialg,I(4),m,n,nhigh,nI,nJ,nK,nL,nlow
   INTEGER          npositives,nsets,nTubes,n_at_OccurProbMax,c(4),SUMr
   CHARACTER*1      Fileno,rise,fall,another_run,another_set,MPNfound
   CHARACTER*9      alg
   CHARACTER*12     Outfile

! Hello Sailor!
   WRITE (*,*)
   WRITE (*,1)

! Set up output file
   WRITE (*,'(/A\)')' Enter a 1 digit output file number:  '
   READ  (*,'(A)') Fileno
   Outfile = 'XactMPN'//Fileno//'.out'
   OPEN (Unit=2,FILE=Outfile)

! Initialise variables
30 DO m = 1,4
    I(m) = 0.D0                                 ! number of tubes in each series
    Vtube(m) = 0.D0                       ! volume of each tube in each series
   ENDDO
   nTubes = 0
   Vtotal = 0.D0

! Input data
31 WRITE (*,'(A\)') ' How many sets of tubes (or wells)? '
   READ (*,*) nsets
   IF (nsets < 1 .OR. nsets > 4) THEN
    WRITE (*,*) ' Can only have 1, 2, 3 or 4 sets'
    GOTO 31
   ENDIF
   WRITE (*,'(A\)') ' Number of tubes in each set, e.g., 1,3,3 (3-tube series)? '
   READ (*,*) (I(m),m=1,nsets)
   WRITE (*,'(A\)') ' mL filtered for, or placed in, each tube in each set ? '
   READ (*,*) (Vtube(m),m=1,nsets)
   CONTINUE

! Number of tubes, volumes and probability of a bacterium's presence in each set
   DO m = 1,nsets
    nTubes = nTubes + I(m)
    Vtotal = Vtotal + I(m)*Vtube(m)
   ENDDO
   pI = I(1)*Vtube(1)/Vtotal
   pJ = I(2)*Vtube(2)/Vtotal
   pK = I(3)*Vtube(3)/Vtotal
   pL = I(4)*Vtube(4)/Vtotal

! Write file and screen headers (with input data echo)
   WRITE (2,*) 'Results for Program XactMPN'
   WRITE (2,*) '=========================='
   IF (nsets == 4) THEN
    WRITE (2,2) (I(m),Vtube(m),m = 1,4),Vtotal
   ELSEIF (nsets == 3) THEN
    WRITE (2,3) (I(m),Vtube(m),m = 1,3),Vtotal
   ELSEIF (nsets == 2) THEN
    WRITE (2,4) (I(m),Vtube(m),m = 1,2),Vtotal
   ELSE
    WRITE (2,5) I(1),Vtube(1),Vtotal
   ENDIF

! Numbers of positives, lower and upper bounds of possible MPN
40 npositives = 0
   IF (nsets == 4) THEN
    WRITE (*,'(A\)') ' Numbers of positive tubes in each set (e.g., 2, 2, 1, 0)? '
   ELSEIF (nsets == 3) THEN
    WRITE (*,'(A\)') ' Numbers of positive tubes in each set (e.g., 2, 2, 1)? '
   ELSEIF (nsets == 2) THEN
    WRITE (*,'(A\)') ' Numbers of positive tubes in each set (e.g., 2, 2)? '
   ELSE
    WRITE (*,'(A\)') ' Number of positive tubes? '
   ENDIF
   READ (*,*) (c(m),m=1,nsets)
   DO m=1,nsets
    IF (c(m) > I(m)) THEN
     WRITE (*,*) 'Cannot have more positives than number of tubes!'
     GOTO 30
    ENDIF
```

```
   ENDDO
   DO m=1,nsets
    npositives = npositives + c(m)
   ENDDO
41 WRITE (*,'(A\)') ' Lower and upper bounds for possible MPN (/100 mL) = ? '
   READ (*,*) MPNlow,MPNhigh
   nlow  = INT(MPNlow *Vtotal/100.D0)
   nhigh = INT(MPNhigh*Vtotal/100.D0) + 1                    ! "+ 1" to round up
   WRITE (*,'(/A)') ' RESULTS'

! Write sub-header for each set of results on file
   IF (nsets == 4) THEN
    WRITE (2,6) (c(m),m=1,4)
   ELSEIF (nsets == 3) THEN
    WRITE (2,7) (c(m),m=1,3)
   ELSEIF (nsets == 2) THEN
    WRITE (2,8) (c(m),m=1,2)
   ELSE
    WRITE (2,9) c(1)
   ENDIF

! Special cases (sterile; all positive)
   SUMr = 0
   DO m = 1,nsets
    SUMr = SUMr + c(m)
   ENDDO
   IF (SUMr == 0) THEN
    WRITE (*,'(A/)') ' Sterile tubes; no bacteria found'
    WRITE (2,'(A/)') ' Sterile tubes; no bacteria found'
    GOTO 30
   ENDIF
   IF (SUMr == nTubes) THEN
    WRITE (*,*)'All tubes positive; can''t tell you a concentration.'
    WRITE (*,10)
    WRITE (*,*)
    WRITE (2,*)'All tubes positive; can''t tell you a concentration.'
    WRITE (2,10)
    WRITE (2,*)
    GOTO 40
   ENDIF

! Now do the main calculation loop. To avoid numerical overflows, calculate
! the entire summed term in (1), (2) or (3) as the exponentiation of
! logarithms; in that term, factorials can get very big while the probabilities
! Pr(r|R,nR) and powers of volume proportions can get very small--and they say
! that harm can come to a young man like that!. [No summing is required for the
! single-set case (eq. 4) of course.] To calculate those  probabilities we may
! use two algorithms succesively: direct ("ialg"=1) or recursive ("ialg"=2).
! These are presented and discussed in the FUNCTION "PrrRn" below. The former
! can be much faster but is slightly noisy at low occurrence probabilities
! (~10^-6) and so can confound the mode selection process. So we first use the
! direct algorithm, but if the maximum calculated occurrence probability is
! less than a prescribed threshold ("TrustPr") we recalculate using the slower
! recursive algorithm.

! Threshold of trustworthy probabilities for calculation of Pr(r|R,n) by the
! direct algorithm, and default algorithm
   TrustPr = 1.D-6
   alg = '   direct'

! Start main calculation
50 rise = 'N'           ! will be set to 'Y' if successive probabilities rise
   fall = 'N'           ! will be set to 'Y' if successive probabilities fall
   MPNfound = 'N'           ! may be set to 'Y' if rise and fall are both 'Y'
   OldOccurProb = 0.D0
   OccurProbMax  = 0.D0
   n_at_OccurProbMax = 0
   DO n = nlow, nhigh
    OccurProb = 0.D0
    IF (alg == 'recursive') THEN
     ialg = 2
    ELSE
     ialg = 1
    ENDIF
    IF (nsets == 4) THEN                                      ! four tube case
     DO nI = 0, n
      DO nJ = 0, n-nI
       DO nK = 0, n - (nI + nJ)
        nL = n - (nI + nJ + nK)
        zeta = PrrRn(c(1),I(1),nI,ialg)*PrrRn(c(2),I(2),nJ,ialg)
        zeta = PrrRn(c(3),I(3),nK,ialg)*PrrRn(c(4),I(4),nL,ialg)*zeta
        IF (zeta > 1.D-250) THEN
         xi1 = nI*DLOG(pI) + nJ*DLOG(pJ) + nK*DLOG(pK) + nL*DLOG(pL)
         xi2 = FACTLN(nI) + FACTLN(nJ) + FACTLN(nK) + FACTLN(nL)
         xi  = DLOG(zeta) + FACTLN(n) + xi1 - xi2
```

```
         OccurProb = OccurProb + DEXP(xi)
        ENDIF
       ENDDO ! of nK
      ENDDO ! of nJ
     ENDDO ! of nI
    ELSEIF (nsets == 3) THEN                              ! three tube case
     DO nI = 0, n
      DO nJ = 0, n-nI
       nK = n - (nI + nJ)
       zeta = PrrRn(c(1),I(1),nI,ialg)*PrrRn(c(2),I(2),nJ,ialg)
       zeta = PrrRn(c(3),I(3),nK,ialg)*zeta
       IF (zeta > 1.D-250) THEN
        xi = DLOG(zeta) + nI*DLOG(pI) + nJ*DLOG(pJ) + nK*DLOG(pK)
        xi = xi + FACTLN(n) - (FACTLN(nI) + FACTLN(nJ) + FACTLN(nK))
        OccurProb = OccurProb + DEXP(xi)
       ENDIF
      ENDDO ! of nJ
     ENDDO ! of nI
    ELSEIF (nsets == 2) THEN                                ! two tube case
     DO nI = 0, n
      nJ = n - nI
      zeta = PrrRn(c(1),I(1),nI,ialg)*PrrRn(c(2),I(2),nJ,ialg)
      IF (zeta > 1.D-250) THEN
       xi = DLOG(zeta) + nI*DLOG(pI) + nJ*DLOG(pJ)
       xi = xi + FACTLN(n) - (FACTLN(nI) + FACTLN(nJ))
       OccurProb = OccurProb + DEXP(xi)
      ENDIF
     ENDDO ! of nI
    ELSE                                                      ! one tube case
     OccurProb = PrrRn(c(1),I(1),n,ialg)
    ENDIF

! Update MPN if on rising limb
    IF (OccurProb > OccurProbMax) THEN
     OccurProbMax = OccurProb
     MPN_per_100mL = n/(Vtotal/100.D0)                         ! MPN / 100 mL
     MPN_per_L     = n/(Vtotal/1000.D0)                             ! MPN / L
     n_at_OccurProbMax = n
    ENDIF

! Write intermediate results to screen and file; update MPN if on rising limb
    IF (nsets == 4) THEN
     WRITE (*,11) c(1),c(2),c(3),c(4),n,OccurProb,n/(Vtotal/100.D0)
     WRITE (2,11) c(1),c(2),c(3),c(4),n,OccurProb,n/(Vtotal/100.D0)
     ELSEIF (nsets == 3) THEN
     WRITE (*,12) c(1),c(2),c(3),n,OccurProb,n/(Vtotal/100.D0)
     WRITE (2,12) c(1),c(2),c(3),n,OccurProb,n/(Vtotal/100.D0)
    ELSEIF (nsets == 2) THEN
     WRITE (*,13) c(1),c(2),n,OccurProb,n/(Vtotal/100.D0)
     WRITE (2,13) c(1),c(2),n,OccurProb,n/(Vtotal/100.D0)
    ELSE
     WRITE (*,14) c(1),n,OccurProb,n/(Vtotal/100.D0)
     WRITE (2,14) c(1),n,OccurProb,n/(Vtotal/100.D0)
    ENDIF

! Check for rise or fall (used later to find the mode)
    IF (n > nlow) THEN
     IF (OldOccurProb <= OccurProb) THEN
      rise = 'Y'
     ELSE
      fall = 'Y'
     ENDIF
    ENDIF
    OldOccurProb = OccurProb

! Finish of n loop
   ENDDO ! of n

! Trustworthy MPN (i.e., mode) found?
   IF (rise == 'Y' .AND. fall == 'Y') THEN
    IF (OccurProbMax > TrustPr) THEN
     MPNfound = 'Y'
    ELSEIF (alg == 'recursive') THEN
     MPNfound = 'Y'
    ENDIF
   ENDIF

! Low probabilities with the direct algorithm? If so, recalculate.
   IF (alg == '   direct' .AND. OccurProbMax <= TrustPr) THEN
    WRITE (*,15) TrustPr
    WRITE (2,15) TrustPr
    alg = 'recursive'
    GOTO 50
   ENDIF
```

```
! Summarise results on the screen and in file
   IF (MPNfound == 'Y') THEN
    WRITE (*,16)
    WRITE (2,16)
    IF (nsets == 4) THEN
     WRITE (*,17) Vtotal,I(1),Vtube(1),I(2),Vtube(2),I(3),Vtube(3),I(4),Vtube(4),         &
                  n_at_OccurProbMax,c(1),c(2),c(3),c(4),OccurProbMax,MPN_per_100mL,MPN_per_L
     WRITE (2,17) Vtotal,I(1),Vtube(1),I(2),Vtube(2),I(3),Vtube(3),I(4),Vtube(4),         &
                  n_at_OccurProbMax,c(1),c(2),c(3),c(4),OccurProbMax,MPN_per_100mL,MPN_per_L
    ELSEIF (nsets == 3) THEN
     WRITE (*,18) Vtotal,I(1),Vtube(1),I(2),Vtube(2),I(3),Vtube(3),n_at_OccurProbMax,      &
                  c(1),c(2),c(3),OccurProbMax,MPN_per_100mL,MPN_per_L
     WRITE (2,18) Vtotal,I(1),Vtube(1),I(2),Vtube(2),I(3),Vtube(3),n_at_OccurProbMax,      &
                  c(1),c(2),c(3),OccurProbMax,MPN_per_100mL,MPN_per_L
    ELSEIF (nsets == 2) THEN
     WRITE (*,19) Vtotal,I(1),Vtube(1),I(2),Vtube(2),n_at_OccurProbMax,                     &
                  c(1),c(2),OccurProbMax,MPN_per_100mL,MPN_per_L
     WRITE (2,19) Vtotal,I(1),Vtube(1),I(2),Vtube(2),n_at_OccurProbMax,                     &
                  c(1),c(2),OccurProbMax,MPN_per_100mL,MPN_per_L
    ELSE
     WRITE (*,20) Vtotal,I(1),Vtube(1),n_at_OccurProbMax,c(1),OccurProbMax,MPN_per_100mL,    &
                  MPN_per_L
     WRITE (2,20) Vtotal,I(1),Vtube(1),n_at_OccurProbMax,c(1),OccurProbMax,MPN_per_100mL,    &
                  MPN_per_L
    ENDIF
    WRITE (*,21)
    WRITE (2,21)
   ELSE
    WRITE (*,*) 'MPN not found in the range requested'
    WRITE (2,*) 'MPN not found in the range requested'
    WRITE (*,*) 'Give me a new range please'
    GOTO 41
   ENDIF

! Another set of results?
   WRITE (*,22) Outfile
   READ (*,'(A)') another_set
   IF (another_set == 'n' .OR. another_set == 'N') GOTO 59
   WRITE (*,*)
   GOTO 40
59 WRITE (*,23)
   READ (*,'(A)') another_run
   IF (another_run == 'n' .OR. another_run == 'N') GOTO 60
   WRITE (*,*)
   GOTO 30
60 CLOSE (UNIT=2)

! Formating the output
1  FORMAT(' ##############################################################################'/  &
          ' #                                                                            #'/  &
          ' #                              "XactMPN"                                     #'/  &
          ' #       Graham McBride, NIWA, Hamilton, New Zealand, May 2002               #'/  &
          ' #                        (g.mcbride@niwa.co.nz)                              #'/  &
          ' #                                                                            #'/  &
          ' #  ========================================================================= #'/  &
          ' #                                                                            #'/  &
          ' # Procedure is based on Tillett & Coleman 1985, J. Appl. Bact. 59: 381-388.  #'/  &
          ' #                                                                            #'/  &
          ' # Input data: output file number                                             #'/  &
          ' #             number of sets of tubes (or wells); maximum is four            #'/  &
          ' #             number of tubes in each set                                    #'/  &
          ' #             mL filtered for, or placed in, each tube in each set           #'/  &
          ' #             number of positive tubes in each set                           #'/  &
          ' #             start and finish values of possible MPNs; you must guess these #'/  &
          ' #                                                                            #'/  &
          ' # The program cycles through a range of possible MPNs, calculating the EXACT #'/  &
          ' # probability of getting each of those numbers distributed in every possible #'/  &
          ' # way among the tubes.  These are shown on the screen and in the results     #'/  &
          ' # file.  If the maximum probability has been found a summary is produced.    #'/  &
          ' # Otherwise you are told that the MPN was not found in the range requested.  #'/  &
          ' # You can step through a number of cases; they will all appear in the output #'/  &
          ' # file.  The theory is described in a NIWA Technical Report 121 ("Preparing  #'/  &
          ' # exact Most Probable Number (MPN) tables using occupancy theory, and        #'/  &
          ' # accompanying measures of uncertainty", 2002), and is also included in the  #'/  &
          ' # comments in this source code. Computations may be very slow in some cases, #'/  &
          ' # e.g., for 5-5-4 positives in a 5x5x5 series. So make an educated guess     #'/  &
          ' # for the start and finish values of possible MPNs.                          #'/  &
          ' #                                                                            #'/  &
          ' # IMPORTANT! THIS CODE CAN BE VERY SLOW FOR SOME EXTREME CASES (E.G., FOR A  #'/  &
          ' # 5x5x5x5 SET TUBES WITH A 5-5-4-1 SET OF POSITIVE RESULTS). YOU MAY NEED TO #'/  &
          ' # ctrl-c TO ABORT SUCH A RUN. (THIS SLOW COMPUTATION PROBLEM HAS YET TO      #'/  &
          ' # BE SOLVED.) IN SUCH A 5-5-x-y CASE (WITH x AND y BETWEEN 0 AND 5) ONE CAN   #'/  &
          ' # GET AN APPROXIMATE ANSWER BY SOLVING THE 5-x-y CASE.                       #'/  &
          ' #                                                                            #'/  &
          ' ##############################################################################')
```

```
2  FORMAT(' Probable numbers of bacteria in 'I2' x 'F6.1' mL, 'I2' x 'F6.2' mL, 'I2' x 'F5.2  &
          ' mL,'/' and 'I2' x 'F6.2' mL.'/                                                     &
          ' Probabilities Pr(i,j,k,l|n) of observing i,j,k,l positive tubes given that '/     &
          ' there were n bacteria in the 'F7.2' mL.')
3  FORMAT(' Probable numbers of bacteria in 'I2' x 'F6.1' mL, 'I2' x 'F6.2' mL, 'I2' x 'F5.2  &
          ' mL.'/' Probabilities Pr(i,j,k|n) of observing i,j,k positive tubes given that '/  &
          ' there were n bacteria in the 'F7.2' mL.')
4  FORMAT(' Probable numbers of bacteria in 'I2' x 'F6.1' mL and 'I2' x 'F6.2' mL.'/          &
          ' Probabilities Pr(i,j|n) of observing i,j positive tubes given that there'/       &
          ' were n bacteria in the 'F7.2' mL.')
5  FORMAT(' Probable numbers of bacteria in 'I4' x 'F6.1' mL.'/                               &
          ' Probabilities Pr(i|n) of observing i positive tubes given that there'/           &
          ' were n bacteria in the 'F7.2' mL.')
6  FORMAT(/' (i,j,k,l) = ('I2','I2','I2','I2')'/' =========================')
7  FORMAT(/' (i,j,k) = ('I2','I2','I2')'/' ====================')
8  FORMAT(/' (i,j) = ('I2','I2')'/' ===============')
9  FORMAT(/' (i) = ('I2')'/' ==========')
10 FORMAT(' All we can say is that it will be greater than the highest'/                     &
        & ' calculable concentration, being that obtained when all but'/                    &
        & ' one of the smallest tubes are positive.')
11 FORMAT(' Pr(i='I2', j='I2', k='I2', l='I2' | n= 'I4') = 'D10.4' [= MPN of'F9.2' / 100mL]')
12 FORMAT(' Pr(i='I2', j='I2', k='I2' | n = 'I4') = 'D11.5' [= MPN of' F9.2' / 100mL]')
13 FORMAT(' Pr(i ='I3', j ='I3' | n = 'I4') = 'D11.5' [= MPN of 'F9.2' / 100mL]')
14 FORMAT(' Pr(i ='I3' | n = 'I4') = 'D11.5' [= MPN of 'F9.2' / 100mL]')
15 FORMAT(/' Results not trustworthy using the default (direct) calculation'/                 &
          ' of occurrence probabilities (they are too low, < 'E8.1').'/                      &
          ' They will now be recalculated using the alternative recursive'/                  &
          ' algorithm. This may be rather slower, but it is very accurate,'/                 &
          ' even for such low probabilities.'/)
16 FORMAT(' ###################################RESULTS SUMMARY##########################')
17 FORMAT(' # You have used a total volume of 'F11.2' mL, comprising:                #'/      &
          ' # 'I3' x 'F6.1' mL, 'I3' x 'F6.2' mL, 'I3' x 'F6.2' mL and 'I3' x 'F6.2' mL.  #'/ &
          ' # When there are 'I5' bacteria in that volume the probability of getting #'/      &
          ' # 'I2','I2','I2' and 'I2' positive tubes is maximised (it is ',E12.6').       #'/ &
          ' # So the MPN = 'F12.3' per 100 mL = 'F11.3' per L.               #')
18 FORMAT(' # You have used a total volume of 'F11.2' mL, comprising:                #'/      &
          ' # 'I3' x 'F6.1' mL, 'I3' x 'F6.2' mL, and 'I3' x 'F6.2' mL.                #'/  &
          ' # When there are 'I5' bacteria in that volume the probability of getting #'/      &
          ' # 'I2','I2' and 'I2' positive tubes is maximised (it is 'E12.6').         #'/    &
          ' # So the MPN = 'F12.3' per 100 mL = 'F11.3' per L.               #')
19 FORMAT(' # You have used a total volume of 'F11.2' mL, comprising:                #'/      &
          ' # 'I3' x 'F6.2' mL and 'I3' x 'F6.2' mL.                                 #'/    &
          ' # When there are 'I5' bacteria in that volume the probability of getting #'/      &
          ' # 'I2' and 'I2' positive tubes is maximised (it is 'E12.6').             #'/     &
          ' # So the MPN = 'F12.3' per 100 mL = 'F11.3' per L.               #')
20 FORMAT(' # You have used a total volume of 'F11.2' mL, comprising:                #'/      &
          ' # 'I3' x 'F6.2' mL.                                                     #'/     &
          ' # When there are 'I5' bacteria in that volume the probability of getting #'/      &
          ' # 'I2' positive tubes is maximised (it is 'E12.6').                    #'/      &
          ' # So the MPN = 'F12.3' per 100 mL = 'F11.3' per L.               #')
21 FORMAT(' ###########################################################################'/)
22 FORMAT(' Results stored in file ',A12,';'/' repeat for another set of positives (Y/N)? '\)
23 FORMAT(' Perform calculations for another dilution series (Y/N)? '\)
END


FUNCTION PrrRn(r,bigR,n,ialg)
! Calculates the probability Pr(r|R,n) of getting r out of R positive test
! tubes, where n bacteria are randomly distributed among them--one bacterium
! being sufficient to guarantee a positive tube.  The formula comes from
! occupancy theory (David & Barton, 1962, "Combinatorial Chance", Charles
! Griffin, London, p. 242) and can be most conveniently written as
!
!                  1    R!
!   Pr(r|R,n) = ---- ----- S(n,r)  for r (=< n) = 1, 2,..., R   .............(1)
!                   n
!                  R   (R-r)!
!
! where S(n,r) is the Stirling number of the second kind, defined as
!
!                1   r  ( r        t      n )
!    S(n,r) = --- Sum |  C  (-1)  (r-t)   |  ................................(2)
!                r! t=0 (   t               )

! The upper summation limit may also be taken as r-1, because the summed term
! is zero when t = r.

! This formula flows from Abraham De Moivre's "Doctine of Chances", first
! published in 1718!

! If ialg = 1 the formula (obtained by combining eqs 1 & 2, cancelling the
! common r! term) is solved directly, putting all terms inside the sum (even
! the constants--so that no term gets too large), i.e.,
!
!                  r-1 (     t          R!        ( r-t )n )
```

```
!    Pr(r|R,n) = Sum | (-1)   ---------------- | --- |  |   ..................(3)
!                t=0 (        (R-r)! (r-t)! t! (  R  )  )
!
! Because this series oscillates and can have mixtures of very large and very
! small numbers, we can run into subtractive cancellation problems.  So we
! first sum the positive terms (even t) and negative terms (odd t), and then
! subtract them.  Even so, we still have some subtractions, because we handle
! the ratios in (3) by exponentiating the difference logarithms of terms.
! Also, Lanczos' formula used for the log-factorial terms has tiny errors of
! order 10^(-10) (Press et al. 1986, "Numerical Recipes", CUP).  This can give
! rise to small initial oscillations when n rises above r, giving rise to a
! negative probability.  This is masked in the algorithm below.

! If ialg = 2 a recursive algorithm is used.  This follows from the Stirling
! number's recursive property, that S(n,r) = r.S(r,n-1) + S(n-1,r-1) (see also
! Tillet & Coleman 1985, J. Appl. Bact. 59: 381-388):
!
!    Pr(r|R,n) = alpha.Pr(r|R,n-1) + beta.Pr(r-1|R,n-1)    .................(4)
!
! where alpha = r/R and beta = (R+1-r)/R.

! Care is needed in designing the algorithm that solves (4) in this FUNCTION.
! First note that the computational molecule has the shape below
!
!         | r-1  r
!     ------------   where "o" denotes known old value [of Pr(r|R,n)]
!     n-1 |  o   o   and "n" denotes the unknown new value to be calculated
!       n |      n   from equation 4. The (r-1,n) value is not needed.
!
! We also know that                                                r
!  Pr(0|R,0) = 1
!  Pr(r|R,0) = 0 for r = 1,2,... ,R                      0  1  2  3  4  5  6  7
!  Pr(0|R,n) = 0 for n = 1,2,...                        -----------------------
!  Pr(r|R,n) = 0 for r (=<R) > n                      0 | 1  0  k  k  k  k  k  k
!                                                     1 | 0  x  0  k  k  k  k  k
! So for the r=7, n=11 case we get the matrix         2 | 0  x  x  0  k  k  k  k
! to the right, where:                                3 | 0  x  x  x  0  k  k  k
! "x" denotes a calculated value;                     4 | 0  x  x  x  x  0  k  k
! "k" denotes a known value (zero) that is not     n  5 | k  x  x  x  x  x  0  k
!     used, and                                       6 | k  n  x  x  x  x  x  0
! "n" denotes an unknown value that does not          7 | k  n  n  x  x  x  x  x
!     need to be calculated,                          8 | k  n  n  n  x  x  x  x
!                                                     9 | k  n  n  n  n  x  x  x
!                                                    10 | k  n  n  n  n  n  x  x
!                                                    11 | k  n  n  n  n  n  n  x

! The algorithm is thus based on r loops within n loops (so "old" ==> old n).
! The r and n counters are rr and nn. In each loop nn "Pr_old(rr)" is
! Pr(rr|R,nn-1) and "Pr_new(rr)" is Pr(rr|R,nn). The algorithm is:

!   Set Pr_old(0) = 1, Pr_old(1) = 0 (i.e., first line of matrix)
!   Start nn loop
!     Define rrmin = MAX(1,nn+r-n), rrmax = MIN(r,nn)
!     Start rr loop from rrmin to rrmax
!       Calculate alpha, beta, thence Pr_new(rr)
!     End rr loop
!     Start another rr loop from rrmin to rrmax
!       Set Pr_old(rr) = Pr_new(rr)
!       If rr = 1 and nn =< n-r,  Pr_old(0) = 0
!       If rr = nn and rr < r, Pr_old(rr+1) = 0
!     End rr loop
!   End nn loop

! This FUNCTION returns these values of Pr(r|R,n) for two test cases, compared
! to the super-accurate result obtained from Mathematica(TM):
!      |---------------------------------------------------------------|
!      |algorithm    | r = 6, R = 8, n = 10 |   r = 2, R = 6, n = 12   |
!      |---------------------------------------------------------------|
!      |direct       |  0.4285874963066389  |  0.00002821136454107241  |
!      |recursive    |  0.4285874962806702  |  0.00002821136453764386  |
!      |===============================================================|
!      |Mathematica  |  0.4285874962806702  |  0.00002821136453764388  |
!      |---------------------------------------------------------------|
! This shows that the recursive algorithm is the more accurate of the two.  It
! doesn't involve any subtrations, so has much less rounding error.  The
! direct algorithm is however a lot faster (hugely so in some cases).  But
! the above suggests it may run into trouble if calculated probabilities
! are very small.

! Pr(r|R,n) is abbreviated to PrrRn in this code.

! Specifications
   IMPLICIT NONE
   DOUBLE PRECISION alpha,beta,eta,PrrRn,PrrRn_evens,PrrRn_odds
   DOUBLE PRECISION FACTLN,Pr_old(0:1000),Pr_new(0:1000)
```

```
   DOUBLE PRECISION REALr,REALrr,REALbigR
   INTEGER          bigR,ialg,n,nn,r,rr,rrmin,rrmax,t,tmaxeven,tmaxodd
   REALr = REAL(r); REALbigR = REAL(bigR)

! Error checks
   IF (r > bigR) THEN
    WRITE (*,*) "Oi! more positives than tubes!"
    STOP
   ENDIF
   IF (ialg < 1 .OR. ialg > 2) THEN
    WRITE (*,*) "Oi! what algorithm?"
    STOP
   ENDIF

! RETURN on special cases
   IF (n == 0) THEN                                                     ! n = 0
    IF (r == 0) THEN
     PrrRn = 1.D0                ! certainly no positive tubes if no bacteria
    ELSE
     PrrRn = 0.D0                  ! can't have positive tubes if no bacteria
    ENDIF
    RETURN
   ENDIF
   IF (r == 0 .AND. n > 0) THEN
    PrrRn = 0.D0        ! can't have no positive tubes if bacteria are present
    RETURN
   ENDIF
   IF (r > n) THEN          ! can't have more positive tubes than bacteria
    PrrRn = 0.D0
    RETURN
   ENDIF

! Not a special case, so 1 <= r <= R, n >= 1 and r <= n.  Use the direct
! algorithm (eq. 3, if ialg = 1) or recursive solution (eq. 4, if ialg = 2).
   IF (ialg == 1) THEN
    PrrRn_evens = 0.D0                                ! start of direct algorithm
    PrrRn_odds  = 0.D0
    IF (MOD(r,2) == 0) THEN                                         ! r is even
     tmaxeven = r-2                       ! upper summation bound for +ve terms
     tmaxodd  = r-1                       ! upper summation bound for -ve terms
    ELSE                                                              ! r is odd
     tmaxeven = r-1                       ! upper summation bound for +ve terms
     tmaxodd  = r-2                       ! upper summation bound for -ve terms
    ENDIF
    DO t = 0,tmaxeven,2
     eta = FACTLN(bigR) + n*DLOG((REALr-REAL(t))/REALbigR)
     eta = eta - (FACTLN(bigR-r) + FACTLN(t) + FACTLN(r-t))
     PrrRn_evens = PrrRn_evens + DEXP(eta)
    ENDDO
    DO t = 1,tmaxodd,2
     eta = FACTLN(bigR) + n*DLOG((REALr-REAL(t))/REALbigR)
     eta = eta - (FACTLN(bigR-r) + FACTLN(t) + FACTLN(r-t))
     PrrRn_odds  = PrrRn_odds  + DEXP(eta)
    ENDDO
    PrrRn = PrrRn_evens - PrrRn_odds
    IF (PrrRn < 0.D0) PrrRn = 0.D0                       ! mask any -ve result
    RETURN
   ELSE                       ! start of recursive algorithm; "old" is previous n
    Pr_old(0) = 1.D0                 ! This case is n=r=0, which is certainly true
    Pr_old(1) = 0.D0            ! This case is r=1, n=0, which is certainly untrue
    DO nn = 1,n                                     ! recursive loops (r within n)
     rrmin = MAX0(1,nn+r-n)
     rrmax = MIN0(r,nn)
     DO rr = rrmin,rrmax
      REALrr = REAL(rr)
      alpha  = REALrr/REALbigR
      beta   = (REALbigR + 1.D0 - REALrr)/REALbigR
      Pr_new(rr) = alpha*Pr_old(rr) + beta*Pr_old(rr-1)
     ENDDO
     DO rr = rrmin,rrmax
      Pr_old(rr) = Pr_new(rr)
      IF (rr == 1 .AND. nn <= n-r) Pr_old(0)    = 0.D0
      IF (rr == nn .AND. rr < r)  Pr_old(rr+1) = 0.D0
     ENDDO
    ENDDO
    PrrRn = Pr_new(r)
    RETURN
   ENDIF
END


FUNCTION FACTLN(N)
! Returns ln(N!), see Press et al., "Numerical
! Recipes" 1986: 159.  Reprogrammed by G B McBride
! (Water Quality Centre, DSIR, Hamilton, New Zealand) in
```

```
! FORTRAN 77 (no frills).  Checked using D6R4 of the
! Numerical Recipes example book.  Links to GAMMLN only.
!
   IMPLICIT REAL*8 (A-H,O-Z)

   DIMENSION A(100)
   DATA A/100*-1./
   IF (N<0) PAUSE 'negative factorial'
   IF (N<=99) THEN
    IF (A(N+1)<0) A(N+1)=GAMMLN(N+1.D0)
    FACTLN=A(N+1)
   ELSE
    FACTLN=GAMMLN(N+1.D0)
   ENDIF
   RETURN
END


FUNCTION GAMMLN(xx)
! Returns ln[gamma(x)] from Lancos' formula in Press et al. (1986,
! "Numerical Recipes", Cambridge: CUP, p.157]. Algorithm has been
! checked against the Numerical Recipes Example Book (program D6R1).

! Reprogrammed by G B McBride, NIWA (National Institute of Water and
! Atmospheric Research Ltd.), P.O. Box 11-115, Hamilton, New Zealand,
! 10/94.  Ph. +07 8561726, fax +07 8560151.

! No links with other programs

   IMPLICIT REAL*8 (A-H,O-Z)
   REAL*8 cof(6), stp, half, one, fpf, x, tmp, ser
   DATA cof, stp/76.18009173D0, -86.50532033D0, 24.01409822D0,                             &
                 -1.231739516D0, .120858003D-2, -.536382D-5, 2.50662827465D0/
   DATA half, one, fpf/0.5D0, 1.0D0, 5.5D0/
   x   = xx - one
   tmp = x + fpf
   tmp = (x+half)*LOG(tmp) - tmp
   ser = one
   DO j=1,6
    x   = x   + one
    ser = ser + cof(j)/x
   END DO
   GAMMLN = tmp + LOG(stp*ser)
   RETURN
END
```

Appendix C: Calculating credible intervals

We want to construct an interval such that there is a high probability that the true concentration of micro-organisms lies within a stated range, given the pattern of positive tubes found. These are Bayesian *credible* intervals (or 'Bayesian confidence intervals', Lee 1997),[16] not frequentist *confidence* intervals. Here we describe the two methods for which calculation procedures have been used in this report.

'Naïve' Poisson empirical Bayes intervals

Tillett & Coleman (1985) noted that in principle it is possible to exactly construct an uncertainty interval around the exact MPN using Bayes' rule. That rule gives us such probabilities as

$$\Pr(n|i,j,k) = \frac{\Pr(i,j,k|n)\Pr(n)}{\Pr(i,j,k)} \qquad \text{(C.1)}$$

where the normalising factor is

$$\Pr(i,j,k) = \sum_{n=i+j+k}^{\infty} \Pr(i,j,k|n)\Pr(n) \qquad \text{(C.2)}$$

Credible intervals could then be calculated by selecting lower and upper values of n that cut off a stated tail area (e.g., 0.05 in the case of a 95% interval). However, Tillett & Coleman noted that because we do not have prior knowledge of the distribution of n [i.e., 'Pr(n)', the prior distribution], the calculation 'cannot proceed'. Accordingly, they (and Tillett 1987, 1995) have observed that it may be more appropriate to consider the much narrower MPR (i.e., 'Most Probable Range'), being all values having an occurrence probability exceeding 95% of that obtaining for the MPN. These can be calculated from the **XactMPN** computer program; e.g., for the 1x3x3 *Campylobacter* table they can be read directly off Table 1. This proposal does not appear to have met with favour, as it appears somewhat arbitrary (Beliaeff 1995).

However, calculations based on Equation C.1 *can* proceed if we make some plausible assumptions about the nature of this prior distribution. This has been done by Dalgety (1999) using an Empirical Bayes Method in which the prior distribution is taken as Poisson with mean equal to the exact MPN. For details, see Dalgety (1999).[17]

Note that empirical Bayes methods use the data twice — in the prior and then again in the likelihood — and so their results may be 'overconfident' (Carlin & Lewis 2000, p. 32). Dalgety's approach is a case in point as it ignores the posterior uncertainty about the estimated population parameter (the Poisson mean) and so his intervals are 'naïve' and may be too short (Carlin & Lewis 2000, pp. 78–79). This is an issue worthy of further consideration.

---

[16] Dalgety (1999) has called this a 'Most Probable Range', a usage that conflicts with that proposed by Tillett & Coleman (1985). Dr Judith McWhirter (Department of Statistics, University of Waikato, pers. comm.) has suggested instead the 'Most Credible Range'. In any event it seems desirable to avoid using the term 'confidence' in the descriptor of a Bayesian interval, so as to keep the frequentist/Bayesian distinction clear.

[17] Empirical Bayesian methods were discussed in general by von Mises (1942) with particular application to multiple tube fermentation tests — see also the summary by Lee (1997, sec. 7.8).

Poisson uniform prior method for approximate MPNs

Greenwood & Yule (1917) presented a Bayesian analysis for multiple tube fermentation problems.[18] They assumed that: (i) a uniform prior distribution of positive concentrations up to some limit (i.e., up to that limit all concentrations are equally likely); (ii) microorganisms in the samples are distributed at random according to the Poisson distribution.

Using Greenwood & Yule's terminology we consider the general case of a dilution series N_1 samples of a_1 cc, giving n_1 negative and m_1 positive results; N_2 samples of a_2 cc, giving n_2 negative and m_2 positive results, etc., the probability (P) that the concentration λ does not exceed some value κ is:

$$P = \Pr\left(\lambda \le \kappa \middle| N_i, a_i, n_i\right) = \frac{\int_0^{\kappa} \left[e^{-\lambda a_1 n_1}(1-e^{-\lambda a_1})^{m_1} \cdot e^{-\lambda a_2 n_2}(1-e^{-\lambda a_2})^{m_2} \cdots e^{-\lambda a_n n_n}(1-e^{-\lambda a_n})^{m_n}\right] d\lambda}{\int_0^{w} \left[e^{-\lambda a_1 n_1}(1-e^{-\lambda a_1})^{m_1} \cdot e^{-\lambda a_2 n_2}(1-e^{-\lambda a_2})^{m_2} \cdots e^{-\lambda a_n n_n}(1-e^{-\lambda a_n})^{m_n}\right] d\lambda} \qquad \text{(C.3)}$$

where w is the length of the uniform prior and $m_i = N_i - n_i$. This equation is based on the probability densities $p(m_i \text{ positive}, n_i \text{ negative} \mid \lambda, a_i) = {}^{N_i}C_{n_i}\left[e^{-\lambda a_i n_i}\left(1-e^{-\lambda a_i}\right)^{m_i}\right]$.

We can evaluate Equation C.3 numerically by writing it as

$$\Pr\left(\lambda \le \kappa \middle| N_i, a_i, n_i\right) = \frac{\Psi(0,\kappa)}{\Psi(0,w)} = \frac{\Psi(0,\kappa)}{\Psi(0,\kappa)+\Psi(\kappa,w)} \qquad \text{(C.4)}$$

where[19]

$$\Psi(a,b) = \int_a^b \psi(\lambda)d\lambda \quad \text{with density function} \quad \psi(\lambda) = \prod_{i=1}^{n_{\text{sets}}} e^{-\lambda a_i n_i}\left(1-e^{-\lambda a_i}\right)^{m_i} \qquad \text{(C.5)}$$

where n_{sets} is the number of dilution series and w is taken to be sufficiently large to ensure that $\partial\Psi(\kappa,w)/\partial w \to 0$, i.e., $\Pr(\lambda \le \kappa|\bullet)$ is constant. Integration of $\psi(\lambda)$ in the $\Psi(a, b)$ term has been effected in a Fortran code (**Greenwood_Yule_MPN**) using ten-point Gauss-Legendre quadrature.[20] This code is listed in Appendix D.

The MPN is defined as the mode of the frequency distribution. Noting that w is chosen to ensure that $\Psi(0, w)$ is a constant, the MPN is therefore the value of λ where $d\psi(\lambda)/d\lambda = 0$. The resulting equation is rather messy, but is made much simpler (Greenwood & Yule 1917) by normalising by ψ, so that

$$z = \frac{1}{\psi}\frac{d\psi(\lambda)}{d\lambda} = \sum_{i=1}^{n_{\text{sets}}}\left[-a_i n_i + \frac{a_i m_i e^{-\lambda a_i}}{1-e^{-\lambda a_i}}\right] = 0 \qquad \text{(C.6)}$$

---

[18] The 'classical' frequentist methods had not yet been developed in the statistical literature! Their paper's opening sentence is 'We were recently consulted by an officer serving on the Western Front as to the significance attaching to ordinary bacteriological methods of gauging the potability of waters'. It then proceeds through an analysis of conflicting standard tables, via an erudite Bayesian analysis.

[19] To compare our nomenclature with that of Greenwood & Yule note that $\Psi(0, w) = 1/y_0$ and $\psi = y/y_0$.

[20] The algorithm is based on subroutine 'qgaus' in Press et al. (1992: 141–142), but uses the more accurate table of normalised weights (w) and abscissae (x) given by Abramowitz & Stegun (1972: table 25.4), i.e.,
w_i: 0.295524224714753, 0.269266719309996, 0.219086362515982, 0.149451349150581, 0.066671344308688
x_i: 0.148874338981631, 0.433395394129247, 0.679409568299024, 0.865063366688985, 0.973906528517172

This equation has been solved for the required value of λ using the standard Newton-Raphson iterative approach (e.g., Dorn & McCracken 1972),

$$\lambda_j = \lambda_{j-1} - \frac{z(\lambda_{j-1})}{\left(dz/d\lambda\right)\big|_{\lambda_{j-1}}} \tag{C.7}$$

until successive differences ($\lambda_j - \lambda_{j-1}$) are 'small'[21], noting that the required derivative is given by

$$\frac{dz}{d\lambda} = \sum_{i=1}^{n_{sets}} -a_i^2 m_i \frac{e^{-\lambda a_i}}{\left(1-e^{-\lambda a_i}\right)^2} \tag{C.8}$$

Note that, using more approximate methods, Greenwood & Yule got MPN = 0.965 and 0.838 per L for 30-5-2 and 21-6-4 positives in a series of 338 x (100cc, 10cc and 1 cc).[22] Given the computational procedures of the day — rooms full of (people) computers — that's an impressive result; using **XactMPN** for a 338x338 x338 setup we get 1.04 and 0.826 / L.

Central and noncentral credible intervals

As is the norm for 100(1–α)% confidence interval limits, one may calculate central credible interval limits such that there is an area of no more than ½α in each tail of the probability density (or mass) function.[23] This is called a 'central credible interval'. These are simply, if somewhat laboriously, calculated using the probability functions given above.[24]

However, if the probability density (or mass) function is asymmetrical such an equal-tailed interval is not the shortest possible interval. This interval may be constructed by relaxing the requirement of having an area of no more than ½α in each tail, by merely requiring that the total tail area cut off by the interval limits is no more than α. For smooth unimodal distributions such an interval has two desirable properties (Lee 1997, p. 49): (i) the probability density (or mass) at any point inside the interval is greater than that at any point outside the interval; (ii) the interval is the shortest of all possible intervals cutting off a total tail area of α. In that case the interval is known as the HPDR (Highest Posterior Density Region). The HPDR has another desirable property that greatly assists its calculation: (iii) the densities at the upper and lower limits are equal.[25]

Accordingly, for the Greenwood & Yule approach we have also calculated the HPDR, again using the Newton-Raphson procedure to obtain values for the abscissae of the density function, noting that its derivative is given simply by

---

[21] Taken as 10^{-12} in these calculations.

[22] In section V(*e*) of their paper; they also had 0.1 cc and 0.01 cc series, but they were all 0 and their approximate computational method ignores zero results.

[23] The discrete case, as when we calculate exact MPNs, gives rise to a probability *mass* function, in which case the required tail areas will nearly always be *less than* ½α. But the continuous case (λ is continuous in the Greenwood & Yule approach), we obtain a probability *density* function, in which case the tail areas can be made *equal to* ½α.

[24] For the **Greenwood_Yule_MPN** program a square root relaxation procedure has been used, as explained in the comments in the 'CL' Function in the Fortran code listed in Appendix D.

[25] Casella & Berger (1990, p. 430) demonstrated the following theorem. Let $f(x)$ be a unimodal pdf. If the interval [a, b] satisfies: **a.** $\int_a^b f(x)dx = 1-\alpha$; **b.** $f(a) = f(b) > 0$, and **c.** $a \le x^* \le b$, where x^* is a (the?) mode of $f(x)$, then the interval [a, b] is the shortest among all intervals that satisfy **a.**

$$\frac{d\psi(\lambda)}{d\lambda} = \psi z \tag{C.9}$$

HPDRs have not been calculated by Dalgety for the Empirical Bayes approach, but it seems that they will be little different from his central credible intervals as the distributions are very symmetric.

Appendix D: Fortran computer code for Greenwood_Yule_MPN

```
PROGRAM Greenwood_Yule_MPN
! Calculates:
! (a) MPN
! (b) upper and lower limits for the central 100(1-alpha)% level MPN credible interval
! (c) upper and lower limits for the 100(1-alpha)% non-central credible interval , i.e., the
!     Highest Density Posterior Region (HPDR)

! from the Bayesian theory (with uniform prior) put forward by Greenwood & Yule (1917,
! Journal of Hygiene 16: 36-54).

! Uses numerical root-finding (Newton-Raphson) to get the MPN (equation on lines
! 6 & 7 of page 54), and numerical integration (ten-point Gauss-Legendre) with a square-root
! relaxation technique to get the credible interval and HPDR limits.

! BASIC EQUATIONS
! Using the Greenwood & Yule nomenclature we have N(i) samples each of volume a(i)
! giving  n(i) negative and m(i) positive results (i = 1, 2,... , imax). Then the probability
! (P) of the concentration (lambda) being within the range [0, kappa] is
!
!    P = Pr[lambda=<kappa | N(i),a(i),m(i),n(i)] = y0[integral of y: lambda = 0 to kappa] .(1)
!
! where n(i) = N(i) - m(i), and the integrand is
!
!    y = y0.phi ..........................................................................(2)
!
! where
!
!                  n_sets
!   phi(lambda) = Product {exp[-lambda.a(i).n(i)][1-exp[-lambda.a(i)]^m(i)]} .............(3)
!                   i=1
!
! where "nsets" is the number of dliution series and the normalising constant is
!                                                     -1
!    y0 = [integral of phi: lambda = 0 to w]  ...........................................(4)
!
! where w --> infinity (taken as the value of lambda at which successive values of 1/y0
! differ by no more than "errortol").  To get the MPN we need the derivative of y (i.e., the
! second derivative of P), set to zero. This derivative is most conveniently taken as (see
! Greenwood & Yule 1917, p. 54)
!
!                1   d(y)
!    z(lambda) = - --------- .............................................................(5)
!                y d(lambda)
!
! so that
!
!                n_sets [                a(i)m(i)exp(-lambda.a(i)) ]
!    z(lambda) =  Sum   | -a(i)n(i) +  ------------------------- | ........................(6)
!                 i=1   [                 1 - exp(-lambda.a(i))    ]
!
!
! and so
!
!       d(z)    n_sets [       2         exp[-lambda.a(i)]     ]
!    -------- =  Sum   | -a(i)  m(i) ----------------------- | ............................(7)
!    d(lambda)   i=1   |                                    2 |
!                      [             [1 - exp(-lambda.a(i)]  ]
!
! and
!
!     d(phi)
!   --------- = phi.z ....................................................................(8)
!   d(lambda)

! We take imax = 5, as Greenwood & Yule present results for a 5 series of dilutions (each with
! 338 tubes!). I have never seen a report from a six-or-more tube series!

! The MPN is obtained by setting z (eq. 6) to zero and solving for the root using
! Newton-Raphson (in FUNCTION MPN, calling FUNCTIONS z and zdot to evaluate eqs. 6 and 7).

! The central credible interval limits are obtained sequentially by guessing a value of phi and
! using another Newton-Raphson procedure (in FUNCTION "Limit", calling functions "phi" and "z"
! to evaluate eqs. 3 and 8) to return the associated abscissa (i.e., limit) and then iterating
! (in FUNCTION "CL", using square root relaxation) so that the tail area cut off by that limit
! is alpha/2, where the tail area is computed by solving eq. 1 using Gauss-Legendre quadrature
! (in FUNCTIONs "P" and "Area"). These limits are returned by the SUBROUTINE "Central_limits".
! (There is also a now-redundant CL FUNCTION ("CL_Redundant"), that uses interval-halving. It
! is too slow but, if called, you will find it returning exactly the same results as its
! replacement.)

! The HPDR limits are obtained simultaneously (in SUBROUTINE "HPDR", also using square root
```

```
! relaxation). This uses the theoretical fact that the limits for the shortest interval
! surrounding a single mode have equal ordinates, hence we obtain the limits simultaneously.

! Programmed in FORTRAN90, by Graham McBride, NIWA, Hamilton, NZ, 4/99, 9/99 5/00, 12/02.
! Fax: 64 7 856-0151; email: g.mcbride@niwa.co.nz

! Results checked against Mathematica.  For a 4-3-1 pattern of positives
! for the 5x10mL, 5x1mL and 5x0.1mL series, Mathematica(TM) gives the
! probability as of not exceeding 77 organisms / 100 mL as
! 0.97555062464344 (at maximum precision).  This program calculates it as
! 0.97555062462487950 (using one interval per unit increase in
! concentration).

! Specifications
   IMPLICIT NONE
   DOUBLE PRECISION a(5), alpha, alpha_percent, confidence_level, errortol, hundred
   DOUBLE PRECISION lower_CL, lower_HPDRL, MPN, MPN_calculated, MPN_guess
   DOUBLE PRECISION Negatives_volume, ninety, one, Total_volume, upper_HPDRL, upper_CL
   DOUBLE PRECISION Vtube(5), zero
   INTEGER          bigN(5), i, m(5), n(5), n_sets, No_positives
   CHARACTER*1      another, Fileno, New_setup
   CHARACTER*10     Outfile
   COMMON /tubes/   a, m, n, n_sets
   COMMON /toler/   errortol

! Numbers
   DATA zero, one, ninety, hundred / 0.D0, 1.D0, 90.D0, 100.D0 /

! Numerical criteria
   errortol = 1.D-12          ! error tolerance in sub-programs (CL, HPDR, Limit, kappa, MPN, P)

! Hello Sailor!
   WRITE (*,*)'###############################################################################'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'#                        "Greenwood_Yule_MPN"                                 #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'#       Based on Greenwood & Yule, 1917, Journal of Hygiene 16: 36-54         #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'#                Graham McBride, NIWA, Hamilton, New Zealand                  #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'#  ========================================================================== #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'# The program calculates three items for a multiple-tube and multiple         #'
   WRITE (*,*)'# dilution series setup                                                       #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'# * MPN (/ 100 mL)                                                            #'
   WRITE (*,*)'# * equi-tailed credible intervals for the MPN (/ 100 mL)                     #'
   WRITE (*,*)'# * Highest Posterior Density Region (HPDR) interval limits (/ 100 mL)        #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'# These are all based on Greenwood & Yule''s Bayesian analysis which uses a    #'
   WRITE (*,*)'# uniform prior distribution and assumes that the tubes constitute a sample   #'
   WRITE (*,*)'# from the water, enabling use of a Poisson approximation. The numerical      #'
   WRITE (*,*)'# techniques used are well documented in the source code--we use a mixture    #'
   WRITE (*,*)'# of Gauss-Legendre quadrature, Newton-Raphson root-finding and a square      #'
   WRITE (*,*)'# root relaxation root-finding procedure (when derivatives do not have        #'
   WRITE (*,*)'# simple closed form). The file is Geenwood_Yule_MPN.f90. Execution time can  #'
   WRITE (*,*)'# be excessive.                                                               #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'# INPUT DATA: output file number;                                             #'
   WRITE (*,*)'#             number of sets of tubes (or wells)--maximum is five;            #'
   WRITE (*,*)'#             number of tubes or wells in each set;                           #'
   WRITE (*,*)'#             mL filtered for, or placed in, each tube or well in each set;   #'
   WRITE (*,*)'#             confidence level for the intervals;                             #'
   WRITE (*,*)'#             number of positive tubes or wells in each set;                  #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'# CHECK:                                                                      #'
   WRITE (*,*)'# For a 3x10 mL, 3x1 mL, 3x0.1 mL setup with 3-2-1 positives you should get   #'
   WRITE (*,*)'#   MPN                           = 149.36          (/ 100 mL)                #'
   WRITE (*,*)'#   equi-tailed credible interval = [55.66, 505.72] (/ 100 mL)                #'
   WRITE (*,*)'#   HPDR                          = [33.71, 448.11] (/ 100 mL)                #'
   WRITE (*,*)'#                                                                             #'
   WRITE (*,*)'###############################################################################'

! Set up output file
   WRITE (*,'(/A\)')' Enter a 1 digit output file number:  '
   READ  (*,'(A)') Fileno
   Outfile = 'GYule'//Fileno//'.out'
   OPEN (Unit=2,FILE=Outfile)

! How many sets of wells (tubes)?
98 New_setup = 'y'
   WRITE (*,'(A\)') ' How many sets of tubes (or wells)? '
   READ (*,*) n_sets
   IF (n_sets < 1 .OR. n_sets > 5) THEN
    WRITE (*,*) ' Can only have 1-5 sets'
```

```
    GOTO 98
   ENDIF

! Initialise tube variables (so no hangovers)
   DO i = 1,5
    a(i) = zero; bigN(i) = 0; m(i) = 0; n(i) = 0; Vtube(i) = zero
   ENDDO

! Get the rest of the data
   WRITE (*,'(A\)') ' Number of tubes in each set--e.g., 1,3,3? '
   READ (*,*) (bigN(i),i=1,n_sets)
   WRITE (*,'(A\)') ' mL filtered for, or placed in, each tube in each set ? '
   READ (*,*) (Vtube(i),i=1,n_sets)
   CONTINUE
   DO i=1,n_sets                                ! Convert tube volumes to proportions of 100 mL
    a(i) = Vtube(i)/hundred
   ENDDO

! What level?
   WRITE (*,'(A\)')' Confidence level, as a percentage (e.g., 95) '
   READ  (*,*) confidence_level
   alpha = one - confidence_level/hundred

! Screen and file headers
   IF (New_setup == 'y') THEN
    WRITE (*,2) confidence_level
    WRITE (2,2) confidence_level
    IF (n_sets == 1) THEN
     WRITE (*,4) bigN(1),Vtube(1)
     WRITE (2,4) bigN(1),Vtube(1)
    ELSEIF (n_sets == 2) THEN
     WRITE (*,5) (bigN(i),Vtube(i),i=1,2)
     WRITE (2,5) (bigN(i),Vtube(i),i=1,2)
    ELSEIF (n_sets == 3) THEN
     WRITE (*,6) (bigN(i),Vtube(i),i=1,3)
     WRITE (2,6) (bigN(i),Vtube(i),i=1,3)
    ELSEIF (n_sets == 4) THEN
     WRITE (*,7) (bigN(i),Vtube(i),i=1,4)
     WRITE (2,7) (bigN(i),Vtube(i),i=1,4)
    ELSEIF (n_sets == 5) THEN
     WRITE (*,8) (bigN(i),Vtube(i),i=1,5)
     WRITE (2,8) (bigN(i),Vtube(i),i=1,5)
    ENDIF
    WRITE (*,10)
    WRITE (2,10)
   ENDIF

! Get the tube results and write to file only
99 WRITE (*,'(A\)') ' Numbers of positive tubes in each set? '
   READ (*,*) (m(i),i=1,n_sets)
   DO i=1,n_sets
    n(i) = bigN(i) - m(i)
   ENDDO
   IF (n_sets == 1) THEN
    WRITE (2,11) (m(i),i=1,n_sets)
   ELSEIF (n_sets == 2) THEN
    WRITE (2,12) (m(i),i=1,n_sets)
   ELSEIF (n_sets == 3) THEN
    WRITE (2,13) (m(i),i=1,n_sets)
   ELSEIF (n_sets == 4) THEN
    WRITE (2,14) (m(i),i=1,n_sets)
   ELSEIF (n_sets == 5) THEN
    WRITE (2,15) (m(i),i=1,n_sets)
   ENDIF

! Guess the MPN (using the approximate formula of Thomas, 1942. JAWWA 34: 572-576). Make the
! MPN guess low (using a multiplier of 90 instead of 100), so that the Newton Raphson
! procedures hunt the peak, not the tail.
   No_positives = 0; Negatives_volume = zero; Total_volume = zero
   DO i = 1,n_sets
    No_positives     = No_positives     + m(i)
    Negatives_volume = Negatives_volume + n(i)*Vtube(i)
    Total_volume     = Total_volume     + bigN(i)*Vtube(i)
   ENDDO
   MPN_guess = ninety*No_positives/DSQRT(Negatives_volume*Total_volume)

! Get MPN and limits
   MPN_calculated = MPN(MPN_guess)
   CALL Central_limits(alpha, MPN_calculated, lower_CL, upper_CL)
   CALL HPDR(alpha, MPN_calculated, lower_CL, upper_CL, lower_HPDRL, upper_HPDRL)

! Write results to screen and file
   WRITE (*,18) MPN_calculated, lower_CL, upper_CL, lower_HPDRL, upper_HPDRL
   WRITE (2,18) MPN_calculated, lower_CL, upper_CL, lower_HPDRL, upper_HPDRL
```

```
! Another run?
   WRITE (*,'(A\)') ' Another set of positives in this setup ?'
   READ (*,'(A)') another
   IF (another == 'n' .OR. another == 'N') GOTO 51
   New_setup = 'n'
   WRITE (*,*)
   GOTO 99
51 WRITE (*,'(A\)') ' Another setup?'
   READ (*,'(A)') another
   IF (another == 'n' .OR. another == 'N') GOTO 50
   WRITE (*,*)
   GOTO 98
50 WRITE (*,30) Outfile
   CLOSE (UNIT=2)

! Formatting the output
2  FORMAT(//' Greenwood-Yule results (/ 100 mL) for confidence level = 'F5.2'%'/&
           ' ================================================================'/)
4  FORMAT(' Single dilution series setup: 'I3' x 'F7.2' mL'/)
5  FORMAT(' Two dilution series setup : 'I3' x 'F7.2' mL; 'I3' x 'F7.2' mL'/)
6  FORMAT(' Triple dilution series setup: 'I3' x 'F7.2' mL; 'I3' x 'F7.2'mL;'/                &
           '                              'I3' x 'F7.2' mL'/)
7  FORMAT(' Four dilution series setup: 'I3' x 'F7.2' mL; 'I3' x 'F7.2' mL;'/                  &
           '                            'I3' x 'F7.2' mL; 'I3' x 'F7.2' mL'/)
8  FORMAT(' Five dilution series setup: 'I3' x 'F7.2' mL; 'I3' x 'F7.2' mL;'/                  &
           '                            'I3' x 'F7.2' mL; 'I3' x 'F7.2' mL;'/                  &
           '                            'I3' x 'F7.2' mL'/)
10 FORMAT(' ---------------------------------------------------------------'/)
11 FORMAT(' Data: 'I3' positives')
12 FORMAT(' Data: 'I3' - 'I3' positives')
13 FORMAT(' Data: 'I3' - 'I3' - 'I3' positives')
14 FORMAT(' Data: 'I3' - 'I3' - 'I3' - 'I3' positives')
15 FORMAT(' Data: 'I3' - 'I3' - 'I3' - 'I3' - 'I3' positives')
18 FORMAT(' MPN = 'F8.2', CR = ['F8.2', 'F8.2'], HPDR = ['F8.2', 'F8.2']')
30 FORMAT(' Results stored in file ',A12)

END


DOUBLE PRECISION FUNCTION MPN(MPN_guess)
! Uses Newton-Raphson method to find the root of the MPN equation (eq 6 in the main program's
! introductory comments). Calls FUNCTIONS z and zdot [for d(z)/d(lambda)]

   IMPLICIT NONE
   DOUBLE PRECISION errortol, MPN_guess, MPN_old, z, zdot
   INTEGER          j
   COMMON /toler/   errortol

! Initialise, DO loop
   MPN_old = MPN_guess
   DO j = 1,100
    MPN = MPN_old - z(MPN_old)/zdot(MPN_old)
    IF (ABS((MPN - MPN_old)/MPN_old) < errortol) GOTO 10
    MPN_old = MPN
   ENDDO
   WRITE (*,*) 'MPN (Newton-Raphson) didn''t converge in 100 iterations'
10 CONTINUE

END


DOUBLE PRECISION FUNCTION z(lambda)
! Returns the normalised derivative of the function y(lambda) for given sets of fermentation
! tubes (see eq. 6 in the main program's introductory comments):
!
!               n_sets [             a(i)m(i)exp(-lambda.a(i)) ]
!   z(lambda) =  Sum   | -a(i)n(i) + ------------------------- |
!                i=1   [               1 - exp(-lambda.a(i))   ]

! Specifications and common variables
   IMPLICIT NONE
   DOUBLE PRECISION a(5), lambda, one, xi, zero
   INTEGER          i, m(5), n(5), n_sets
   COMMON /tubes/   a, m, n, n_sets

! Numbers
   DATA zero, one / 0.D0, 1.D0 /

! Initialise, DO loop
   z = zero
   DO i = 1, n_sets
    xi    = DEXP(-lambda*a(i))
    z     = z - a(i)*(n(i) - m(i)*xi/(one - xi))
   ENDDO
```

```
END


DOUBLE PRECISION FUNCTION zdot(lambda)
! Returns the derivative of the function z(lambda) for given sets of fermentation tubes
! (see eq. 7  in the main program's introductory comments):
!
!                   n_sets [       2        exp[-lambda.a(i)]    ]
!    zdot(lambda)  = Sum   | -a(i)  m(i) ----------------------- |
!                    i=1   |                                   2 |
!                          [             [1 - exp(-lambda.a(i)]  ]

! Specifications and common variables
   IMPLICIT NONE
   DOUBLE PRECISION a(5), lambda, one, xi, zero
   INTEGER          i, m(5), n(5), n_sets
   COMMON /tubes/   a, m, n, n_sets

! Numbers
   DATA zero, one / 0.D0, 1.D0 /

! Initialise, DO loop
   zdot = zero
   DO i = 1, n_sets
    xi   = DEXP(-lambda*a(i))
    zdot = zdot - a(i)*a(i)*m(i)*xi/(one - xi)**2
   ENDDO

END


SUBROUTINE Central_limits(alpha, MPN_calculated, lower_CL, upper_CL)
! Returns lower and upper alpha-level limits for central credible interval on MPN.
! These cut off an area of alpha/2 in each tail of the posterior distribution.

! Specifications
   IMPLICIT NONE
   DOUBLE PRECISION alpha, CL, lower_CL, MPN_calculated, upper_CL

! Limits
   lower_CL = CL(alpha, MPN_calculated, 'lower')
   upper_CL = CL(alpha, MPN_calculated, 'upper')

END


DOUBLE PRECISION FUNCTION CL(alpha, MPN, which_tail)
! Calculates either the lower or the upper 100(1-alpha)% level limit for the central credible
! interval on MPN. These limits have an area of alpha/2 in each tail of the posterior
! distribution; hence they are central.

! Uses a square root relaxation technique. That is:
! (1) Start with a value of phi that is too high, i.e., calculate the limit corresponding
!     to that phi value and then calculate the tail area it cuts off, making sure that this
!     area is greater than alpha/2.
! (2) If tail area approx equals alpha/2, exit the FUNCTION.
! (3) Multiply phi by a weight (<1), recalculate the limit and recompute tail area
! (4) If tail area > alpha/2, go back to step (2); if not we have overshot and so recover
!     the previous phi value (by dividing by the weight), reset the weight to its square root
!     root and go back to (2)

! Declarations
   IMPLICIT NONE
   DOUBLE PRECISION alpha, critical_tail_area, errortol, half, Limit, MPN, one
   DOUBLE PRECISION P, phi, phi_trial, tail_area, two, weight
   INTEGER          j
   CHARACTER*5      which_tail
   COMMON /toler/   errortol

! Numbers
   DATA half, one, two / 0.5D0, 1.D0, 2.D0 /

! Relaxation weight parameter, desired tail area
   weight = 0.9D0
   critical_tail_area = alpha/two

! Lower CL case, initialise
   IF (which_tail == 'lower') THEN
    CL = MPN/two                                                           ! got a better idea?
    phi_trial = phi(CL)
    tail_area = P(CL)

! Refine, using square root relaxation
    DO j = 1,100
     IF (ABS((tail_area - critical_tail_area)/critical_tail_area) < errortol) RETURN
```

```
10   phi_trial = phi_trial*weight
     CL = Limit(phi_trial, half*MPN)                                 ! hunts below mode (= MPN)
     tail_area = P(CL)
     IF (tail_area < critical_tail_area) THEN
      phi_trial = phi_trial/weight                                     ! recover previous value
      weight = DSQRT(weight)      !adjust weight upward by taking square root, ever closer to 1
      GOTO 10
     ENDIF
    ENDDO
    WRITE (*,*) 'Lower CL didn''t converge in 100 trials'
    RETURN
   ENDIF

! Upper CL case. Initialise
   IF (which_tail == 'upper') THEN
    CL        = MPN*two                                                    ! got a better idea?
    phi_trial = phi(CL)
    tail_area = one - P(CL)

! Refine, using square root relaxation
    DO j = 1,100
     IF (ABS((tail_area - critical_tail_area)/critical_tail_area) < errortol) RETURN
20   phi_trial = phi_trial*weight
     CL = Limit(phi_trial, two*MPN)                                  ! hunts above mode (= MPN)
     tail_area = one - P(CL)
     IF (tail_area < critical_tail_area) THEN
      phi_trial = phi_trial/weight                                    ! recover previous value
      weight = DSQRT(weight)     !adjust weight upward by taking square root, ever closer to 1
      GOTO 20
     ENDIF
    ENDDO
    WRITE (*,*) 'Upper CL didn''t converge in 100 trials '
    RETURN
   ENDIF

END


DOUBLE PRECISION FUNCTION Limit(phi_trial, Limit_guess)
! Returns one of the two possible limits (abscissae) for any value of phi using Newton-Raphson.
! If Limit_guess < MPN get lower limit, else get the upper limit. Uses eq. (8) in the main
! program's introductory comments for the derivative of phi.

   IMPLICIT NONE
   DOUBLE PRECISION Limit_guess, Limit_old, errortol, phi, phi_trial, z
   INTEGER          j
   COMMON /toler/   errortol

! DO loop
   Limit_old = Limit_guess
   DO j = 1,100
    Limit = Limit_old - (phi(Limit_old) - phi_trial)/(phi(Limit_old)*z(Limit_old))
    IF (ABS((Limit - Limit_old)/Limit_old) < errortol) GOTO 10
    Limit_old = Limit
   ENDDO
   WRITE (*,*) 'Limit (Newton-Raphson) didn''t converge in 100 iterations'
10 CONTINUE

END


DOUBLE PRECISION FUNCTION P(kappa)
! Evaluates the Greenwood and Yule integral from lambda = 0 to kappa (their eq. 23).
! This is the posterior probability of getting up to a given concentration limit (kappa),
! under a uniform prior distribution. Calls a Gauss-Legendre routine for each subinterval
! of lambda. Can have a number of subintervals per unit increase in lambda, but trials have
! indicated that n_intervals = 1 works very well (because Gauss-Legendre is so accurate).

! Specifications
   IMPLICIT NONE
   DOUBLE PRECISION a, Area, b,  del_lambda, errortol, kappa, one, phi, w, zero
   DOUBLE PRECISION P_denominator, P_denominator_old, P_numerator
   INTEGER          j, jmax, lambda_warning_limit, kappa_boundary_flag, n_intervals
   CHARACTER*1      infinity_warning
   EXTERNAL         phi                                                     ! phi is a FUNCTION
   COMMON /toler/   errortol

! Numbers
   DATA zero, one / 0.D0, 1.D0 /

! Integration steps
   n_intervals = 1                               ! number of intervals per unit increase in lambda

! Set flags
   lambda_warning_limit = 20000
```

```
   kappa_boundary_flag = 0                          ! kappa is at an interval boundary, else = 1
   infinity_warning = 'n'

! Now do the main calculation, numerator first (a & b are limits of each subintegral)
   del_lambda  = one/n_intervals                                                 ! interval size
   jmax = INT(kappa*n_intervals)                                               ! INT rounds down
   P_numerator = zero
   b = zero                                                                ! initialise for loop
   DO j = 1,jmax                                      ! do up to nearest point to left of kappa
    a = b
    b = b + del_lambda
    P_numerator = P_numerator + Area(phi,a,b)
   ENDDO
   IF (b < kappa) THEN                                  ! still a wee interval to get to kappa?
    kappa_boundary_flag = 1                            ! kappa not at a boundary of an interval
    a = b
    P_numerator = P_numerator + Area(phi,a,kappa)
   ENDIF

! Second part of the denominator (integral from kappa to w). If kappa_boundary_flag = 1 do
! only the second part of the jmax+1 interval, Otherwise do it all. Store the result as the
! start value for the sum.
   j = jmax + 1
   w = j*del_lambda
   IF (kappa_boundary_flag == 1) THEN
    b = b + del_lambda
    P_denominator = Area(phi,kappa,b) + P_numerator
   ELSE
    a = b
    b = b + del_lambda
    P_denominator = Area(phi,a,b) + P_numerator
   ENDIF
   P_denominator_old = P_denominator
   DO j = jmax+2,INT(lambda_warning_limit/del_lambda)
    w = j*del_lambda
    a = b
    b = b + del_lambda
    P_denominator = P_denominator + Area(phi,a,b)
    IF (DABS((P_denominator - P_denominator_old)/P_denominator_old) < errortol) GOTO 75
    P_denominator_old = P_denominator
   ENDDO
   IF (w > lambda_warning_limit .AND. infinity_warning == 'n') THEN
    infinity_warning = 'y'
    WRITE (*,7) lambda_warning_limit
7  FORMAT(' lambda has exceeded 'I5'; calculation will continue (press ctrl-C to stop).')
   ENDIF
75 CONTINUE

! Result
   P = P_numerator/P_denominator

END


DOUBLE PRECISION FUNCTION Area(func,a,b)
! Uses ten-point Gauss-Legendre quadrature to integrate a function ("func") from a to b.
! Based on function qgauss in Press et al. 1992, "Numerical Recipes in Fortran", CUP, but
! with weights taken from Table 25.4 in Abramowitz & Stegun (because they have better
! precision).

! Specifications and data
   IMPLICIT NONE
   DOUBLE PRECISION a, b, func, dx, half, w(5), x(5), xm, xr, zero
   EXTERNAL         func
   INTEGER          j
   SAVE w,x

! Numbers
   DATA zero, half / 0.D0, 0.5D0 /

! Weights
   DATA w/.295524224714753,.269266719309996,.219086362515982,.149451349150581,.066671344308688/
   DATA x/.148874338981631,.433395394129247,.679409568299024,.865063366688985,.973906528517172/
   xm = half*(a+b)

! Algorithm
   xr   = half*(b-a)
   Area = zero
   DO j = 1,5
    dx   = xr*x(j)
    Area = Area + w(j)*(func(xm+dx) + func(xm-dx))
   ENDDO
   Area = xr*Area

END
```

```
DOUBLE PRECISION FUNCTION phi(lambda)
! Returns the function phi for given sets of fermentation tubes (see eq. 3 in the main
! program's introductory comments):
!
!                  n_sets [  -lambda.a(i)n(i) (      -lambda.a(i) ) m(i) ]
!    phi(lambda) =   Pi   | e                 | 1 - e             |      |
!                   i=1   [                   (                   )      ]

! Specifications and common variables
   IMPLICIT NONE
   DOUBLE PRECISION a(5), lambda, one
   INTEGER          i, m(5), n(5), n_sets
   COMMON /tubes/   a, m, n, n_sets

! Number
   DATA one / 1.D0 /

! Initialise, DO loop
   phi  = one
   DO i = 1,n_sets
    phi  = phi * DEXP(-lambda*a(i)*n(i)) * (one - DEXP(-lambda*a(i)))**m(i)
   ENDDO

END


SUBROUTINE HPDR(alpha, MPN, lower_CL, upper_CL, lower_HPDRL, upper_HPDRL)
! Gets HPDR limits, from the property proven by Casella & Berger 1990, "Statistical
! Inference", Wadsworth & Brooks/Cole, at pp. 430-431. That is, let f(x) be a unimodal
! pdf. If the interval [a,b] satisfies
! (a) Integral[f(x) dx, {a,b}] = 1 - alpha
! (b) f(a) = f(b) > 0, and
! (c) a =< x* =< b is the mode of f(x) [C&B say "a" mode!]
! then [a,b] is the shortest of all possible intervals satisfying (a).

! Uses a square root relaxation technique. That is:
! (1) Start with a value of phi that is too high, i.e., calculate the limits corresponding
!     to that phi value and then calculate the total tail area they cut off, making sure
!     that this area is greater than alpha.
! (2) If total tail area approx equals alpha, exit the SUBROUTINE.
! (3) Multiply phi by a weight (<1), recalculate the limits and compute total tail area
! (4) If total tail area > alpha, go back to step (2); if not we have overshot and so
!     recover the previous phi value (by dividing by the weight), reset the weight to its
!     square root and go back to (2). This guarantees an "approach from above".

! Declarations
   IMPLICIT NONE
   DOUBLE PRECISION alpha, errortol, half, Limit, lower_CL, lower_HPDRL, MPN, one, P, phi
   DOUBLE PRECISION phi_trial, total_tail_area, two, upper_CL, upper_HPDRL, weight, zero
   INTEGER          j
   COMMON /toler/   errortol

! Numbers
   DATA half, one, two / 0.5D0, 1.D0, 2.D0 /

! Relaxation weight parameter
   weight = 0.9D0

! Initialise
   phi_trial    = MAX(phi(lower_CL), phi(upper_CL))     ! guarantees that total_tail_area > alpha
   lower_HPDRL = Limit(phi_trial, half*MPN)              ! guarantees trial value < mode (= MPN)
   upper_HPDRL = Limit(phi_trial, two*MPN)               ! guarantees trial value > mode (= MPN)
   total_tail_area = P(lower_HPDRL) + one - P(upper_HPDRL)

! Refine, using square root relaxation
   DO j = 1,100
    IF (ABS((total_tail_area - alpha)/alpha) < errortol) RETURN
10  phi_trial = phi_trial*weight
    lower_HPDRL = Limit(phi_trial, half*MPN)
    upper_HPDRL = Limit(phi_trial, two*MPN)
    total_tail_area = P(lower_HPDRL) + one - P(upper_HPDRL)
    IF (total_tail_area < alpha) THEN
     phi_trial = phi_trial/weight                                       ! recover previous value
     weight = DSQRT(weight)      ! adjust weight upward by taking square root, ever closer to 1
     GOTO 10
    ENDIF
   ENDDO
   WRITE (*,*) 'HPDR didn''t converge in 100 trials'
75 CONTINUE

END


DOUBLE PRECISION FUNCTION CL_Redundant(alpha, MPN, which_tail)
! NOT USED--TOO SLOW--WE USE FUNTION CL INSTEAD. We keep this in the file for reference (it
```

```
! works!). Calculates either the lower or the upper 100(1-alpha)% level limit for the central
! MPN credible interval on MPN. These limits have an area of alpha/2 in each tail of the
! posterior  distribution; hence they are central. Uses interval halving (can't use
! Newton-Raphson because the function doesn't have simple form?)

! Specifications
   DOUBLE PRECISION alpha, CL_Redundant_average, CL_Redundant_guess, CL_Redundant_left
   DOUBLE PRECISION CL_Redundant_right, errortol, MPN, one, tail_area, two
   INTEGER          i
   CHARACTER*5      which_tail
   COMMON /toler/   errortol

! Numbers
   DATA one, two / 1.D0, 2.D0  /

! Define tail area
   IF (which_tail == 'lower') THEN
    tail_area = alpha/two
    CL_Redundant_guess = MPN/two
   ELSE
    tail_area = one - alpha/two
    CL_Redundant_guess = MPN*two
   ENDIF

! A lucky shot?
   IF (P(CL_Redundant_guess) == tail_area) THEN
    CL_Redundant = CL_Redundant_guess
    RETURN
   ENDIF

! Get CL_Redundant values differing by 1 (/100 mL), straddling the tail-area Limit
   IF (P(CL_Redundant_guess) < tail_area) THEN
    CL_Redundant_left  = CL_Redundant_guess
    CL_Redundant_right = CL_Redundant_guess + one

! Straddling already?
    IF(P(CL_Redundant_left) < tail_area .AND. P(CL_Redundant_right) >= tail_area) GOTO 30

! Right CL_Redundant to left of Limit
10  IF(P(CL_Redundant_right) < tail_area) THEN
     CL_Redundant_left  = CL_Redundant_left  + one
     CL_Redundant_right = CL_Redundant_right + one
     GOTO 10
    ELSE
     GOTO 30
    ENDIF
   ELSE
    CL_Redundant_left  = CL_Redundant_guess - one
    CL_Redundant_right = CL_Redundant_guess

! Straddling already?
    IF(P(CL_Redundant_left) <= tail_area .AND. P(CL_Redundant_right) > tail_area) GOTO 30

! Lest CL_Redundant to right of Limit
20  IF(P(CL_Redundant_left) > tail_area) THEN
     CL_Redundant_left  = CL_Redundant_left  - one
     CL_Redundant_right = CL_Redundant_right - one
     GOTO 20
    ELSE
     GOTO 30
    ENDIF
   ENDIF
30 CONTINUE

! CL_left and CL_right now straddle the tail-area's Limit, so we can start the
! interval-halving procedure, which is
! (a) Calculate P with current values of CL_left and CL_right.
! (b) Calculate CL_average, where CL_average = (CL_left + CL_right)/2.
! (c) Check for convergence, by three criteria
!      * if |P(CL_left) - tail_area|    < errortol, then CL = CL_left,    exit FUNCTION
!      * if |P(CL_right) - tail_area|   < errortol, then CL = CL_right,   exit FUNCTION
!      * if |P(CL_average) - tail_area| < errortol, then CL = CL_average, exit FUNCTION
! (d) If P(CL_average) < tail_area then replace CL_right by CL_average,
!     if P(CL_average) > tail_area then replace CL_left  by CL_average,
! (e) Go to (a), for a maximum of 1000 times (if no convergence by then, "trouble at mill")
   DO i = 1,100
    CL_Redundant_average = (CL_Redundant_left + CL_Redundant_right)/two
    IF (DABS(P(CL_Redundant_left) - tail_area) < errortol) THEN
     CL_Redundant = CL_Redundant_left
     RETURN
    ENDIF
    IF (DABS(P(CL_Redundant_right) - tail_area) < errortol) THEN
     CL_Redundant = CL_Redundant_right
     RETURN
    ENDIF
```

```
    IF (DABS(P(CL_Redundant_average) - tail_area) < errortol) THEN
     CL_Redundant = CL_Redundant_average
     RETURN
    ENDIF
    IF (P(CL_Redundant_average) < tail_area) THEN
     CL_Redundant_left  = CL_Redundant_average
    ELSE
     CL_Redundant_right = CL_Redundant_average
    ENDIF
   ENDDO
   WRITE (*,*) 'CL_Redundant function did not converge in 100 iterations!'

END
```

NIWA
Taihoro Nukurangi

National Institute of
Water & Atmospheric
Research Ltd