Programming challenges of sampling controls to cases from the dynamic risk sets in nested case–control studies

Victor A Kiri[1,2]

[1]FV&JK Consulting Ltd, Surrey, UK, [2]Centre of Biostatistics, University of Limerick, Ireland

Pharmacoepidemiological studies based on the cohort design are simpler to analyse and their results easier to interpret. However, these may not reflect real-life drug use which is a major strength of such studies. The nested case–control design is often used instead to avoid the computational burden associated with time-dependent explanatory variables. Unlike the classical case–control design which is generally easy to programme, that of the nested case–control can pose a number of challenges. Subjects can be chosen as controls more than once and a subject who is chosen as a control can later become a case. Indeed controls are chosen from among those in the cohort who are at risk of the event at that time (i.e. we sample from the risk set defined by the case). We highlight the main programming challenges of the design as well as describe and demonstrate approaches for resolution and appropriate implementation.

Keywords: Pharmacoepidemiological studies, Confounding, Incidence density sampling, Time-dependent effect, Conditional logistic regression

Introduction

The randomized controlled trial is universally considered as gold standard in drug research primarily because it is capable of providing the most compelling evidence on the efficacy of a healthcare intervention.[1] In a clinical trial, we conceal treatment allocation by a randomization process that we maintain through blinding to study participants. However, due to cost implications and other logistical considerations, features such as small sample size, short study duration, and restricted populations often make trials inadequate for determining rates of adverse effects because of low frequencies. Today, many consider the post-marketing phase as the richest source of safety data and clinical trials constitute only a part of drug safety research. Pharmacoepidemiological studies on drug safety are becoming increasingly popular mainly because they reflect real-life utility of drugs, although evidence from these studies is often dismissed because of the absence of randomization and the likely consequences of selection bias due to confounding. Confounding is the problem that arises when the effect of a factor that is associated with both the outcome of interest and exposure is mixed up with the actual exposure effect.

Part of the reason for the increasing growth in these studies is because in some situations, a trial may be either inappropriate or inadequate or even impossible to conduct-making pharmacoepidemiological study the only alternative to doing nothing. The other is perhaps, because when properly conducted, these studies are capable of providing compelling information about the safety of drugs in certain populations and settings.

Cohort studies are by far the most common in epidemiological research. Although cohort studies which are based on intention-to-treat approach offer a simple design with data which are simpler to analyse and results easier to interpret, such studies also intrinsically assume that any time-varying treatment effect that exits can be adequately estimated by a fixed-effect component. However, such an assumption may not reflect real-life drug use. Indeed, where sophisticated models such as those involving time-varying components are used, these are usually based on assumptions which may be difficult to satisfy and their results similarly difficult to interpret. Another drawback of the cohort design for investigating the causes of disease with low incidence is that large and lengthy studies may be required to command adequate statistical power.

The case–control design is an alternative which avoids this difficulty. In the cohort design, we follow the entire cohort and measure the rate of occurrence of new cases in the different exposure groups to investigate possible relationship between exposure and disease incidence. The follow-up allows us to

Correspondence to: V A Kiri, FV&JK Consulting Ltd, Guildford, Surrey GU1 1NX, UK. Email: Victor.Kiri@fvjkconsult.com

Published by Maney
DOI 10.1179/1757092112Z.0000000004

identify those subjects who develop the disease during the study period as well as those who remain free of the disease. By contrast, in a classical case–control design, we identify the subjects who develop the disease (the cases) by some mechanism other than that of follow-up and a group of healthy subjects (the controls) is used to represent the subjects who do not develop the disease. Here, we try to control for the potential effects of confounding factors by matching on such covariates in preference over modelling the covariate effects. From programming perspective, the main challenge effectively depends on the size of the control population as well as the sampling procedure used to obtain the controls matched to the cases.

Studies based on the case–control design can be conducted and completed in a shorter time. Such studies generally require smaller sample sizes and hence are usually less expensive. In general, it is the only practical approach for identifying risk factors for rare diseases, especially where follow-up of a large sample for occurrence of the condition might be impractical. The design is particularly useful for reducing the labour and cost of data collection, assessing an exposure not measured in the original cohort and for avoiding the computational burden associated with multiple time-dependent explanatory variables.

In the case–control design, the period over which failures are registered as cases is called the study period and it is theoretically possible to choose controls to represent the entire source population at the start of the study period, including those who go on to become cases. However, this is rarely achievable by the classical approach, because the source population is often ill-defined especially in routinely collected databases such as those based on healthcare claims. The nested case–control version is used to overcome this problem as the controls that are matched to each case are obtained by sampling from the very cohort that gave rise to the cases. In other words, the controls for each case are chosen from among those in the cohort who are at risk of the event at that time (i.e. we sample from the risk set defined by the case). Intuitively, comparison of exposure between the case and controls in the risk sets does not require all the controls; a representative sample of the controls should be sufficient. Subjects can be chosen as controls more than once, and a subject who is chosen as a control can later become a case. Indeed, most safety-focused studies are based on this design so as to avoid the computational burden associated with time-dependent explanatory variables. The Cox regression is used to estimate the relative change in event rates per unit of increase in exposure, in which the case–control comparison from each risk set is quantified in a conditional logistic partial likelihood contribution.

The strength of the nested case–control design rests largely on the appropriateness of the controls matched to the cases, which can be quite daunting from programming perspective. We need to sample the controls from the dynamic risk sets while ensuring the desired representative quality. We will highlight the main programming challenges of the design as well as describe and demonstrate approaches for resolution and appropriate implementation.

Methods

There are various methods for sampling controls in nested case–control analyses, but the method of incidence density sampling has been identified as the method of choice for obtaining unbiased results.[2] An efficient incidence density sampling scheme is one in which controls are selected without replacement from all persons at risk at the time of case occurrence, excluding the index case itself.[3]

To use incidence density sampling to match controls to cases, the programmer must provide a data set (i.e. the 'source') that describes the study cohort and includes the relevant variables such as those that define the case and treatment exposure statuses of patients as well as those for matching.[4] Although some programs for incidence density sampling have been previously described elsewhere,[5–7] none of them is based on the SAS PROC SQL which offers a robust but simple approach to matching of patients on pre-defined variables.

As in the classical case–control study setting, data from a nested case–control study involve series of standard 2×2 contingency tables — each table being identified in SAS as a stratum, composing of four cells that represent the frequencies of cases and controls classified by exposure status in terms of whether each patient in the stratum is exposed or unexposed as shown in Table 1.

In a typical nested case–control study involving incidence density sampling, each case will be matched to n controls ($n \geqslant 1$). Thus, in the matched dataset, if $a=1$ (i.e. the case is exposed), then $b=0$ and *vice versa* for each stratum. Matching is usually on categorical variables such as age, gender, etc. and in general, the more variables that are involved in the matching of cases to controls, the smaller the matched study sample size.

Programming approach

Based on a cohort source dataset which we identify as '***fullset***', our program uses SAS PROC SQL to match

Table 1 Illustration of a standard 2×2 contingency table

	Exposed	Unexposed
Case	*a*	*b*
Control(s)	*c*	*d*

```
* The Matching procedure- matching each pateid case to every qualified pateid
on ageindex and gender *;
proc sql;
create table join1 as select distinct
   a.pateid_case, a.tcens, a.ageindex, a.gender,
   b.pateid, ranuni(12345) as rand_num
   from cases a inner join fullset b
   on (a.tcens) <=b.tcens       /* control is in risk set at a.tcens */
   and a.pateid case ne b.pateid  /* control is not the case */
   and a.ageindex-1 <= b.ageindex <= a.ageindex+1  /* similar age +-1 */
   and a.gender = b.gender          /* control is same gender as case */
   order by pateid case, rand num;
quit;

* Macro that selects only "n" matched controls for each case *;
%Macro match(controlsnum);
data match(keep=pateid case pateid);
  set join1;
  by pateid case rand num;
  retain mcount;
  mcount+1;
  if first.pateid_case then mcount=1;
  if mcount le &controlsnum;
run;

%mend match;
quit;

* Input the number of controls (n) to match to each case *;
%match(n);


* Output full dataset of matched cases and controls *;
data matched_cases_controls_all;
  set match;
run;

proc printto;
run;



*To separate the matched cases from their matched controls for purpose of
assigning event flag and also for obtaining the covariates in the database *;
data matched_cases_controls;
  merge matched cases controls all(in=a)
        pnsurv.study all(in=b keep=pateid tcens
                                    rename=(pateid=pateid case));
  by pateid case;
  if a=1 and b=1;
run;

proc sort data=matched cases controls;
by pateid case;
run;
```

Figure 1 Program for matching cases to controls based on incidence density sampling

```
* To retain the matching identifier (i.e. assign case identifier to its
matched controls) for accurate update of the covariates at corresponding
event time and assignment of censoring status *;

data case_id (keep=pateid pateid_case tcens);
  set matched cases controls;
  by pateid case;
  if first.pateid_case; /* only the first observation to identify a case */
        pateid=pateid case; /* make the case id as the identifier */
run;

data id (keep= pateid pateid_case matchset tcens); /*generate id to link
cases with their matched sets */
set case id;
by pateid_case;
matchset= N ;
run;

proc sort data=matched_cases_controls;
by pateid;
run;

proc sort data=case_id;
by pateid;
run;

* To obtain follow-up time and if needed, generate updates of covariates for
each matched control *;
data matched_cases_controls1;
merge matched_cases_controls case_id(keep= pateid pateid_case tcens);
by pateid pateid case;
run;

proc sort data=matched_cases_controls1;
by pateid_case;
run;

data final_matched_temp; /* Assign the matchset identifier to all records */
merge matched cases controls1 id(keep=pateid case tcens matchset
rename=(matchset=statu));
by pateid case;
run;

proc sort data= final_matched_temp; /* reordered by original identifier */
by pateid;
run;

data pnsurv.final_matched_all(drop=pateid_case); /* ready for analysis */
set final matched temp;
if pateid=pateid_case then casecon=1;
else casecon=0;
run;
```

Figure 1 Continued

each case to all possible controls (Fig. 1). For successful implementation, the following four variables are vital: *pateid* (unique patient identifier), *tcens* (follow-up time), *ageindex*, and *gender* — the last two as matching variables. As first step, a separate file is created out of '***fullset***' comprising of only the patients who experienced the event of interest ('***cases***' with '*pateid*' renamed as '*pateid_case*') at any time during the study period. The INNER JOIN option ensures that any person from '***fullset***' that could match to more than one case from '***cases***' gets linked to all possible cases for which the person qualifies as a potential control and also removes those that are not matched to any case. The '$a.tcens \leqslant b.tcens$' condition ensures that the time at risk is at least as long for the control (*b.pateid*) as it is for the case (*a.pateid_case*),

whereas the condition '*a.pateid_case ne b.pateid*' ensures that a patient does not get matched to itself. The variable '*rand_num*' is used to choose random controls from the possible controls. The macro **%Macro** is used to ensure that no more than *n* controls are matched to each case. Lastly, the variable casecon defines the case status of patients in the matched dataset with casecon=1 for a case.

The **%Macro** is followed by six datasteps — primarily to construct the individual 2×2 matrices (i.e. Table 1) which will involve separating the matched cases from their matched controls, assigning censoring status flag to each case and control ('*casecon*'=1 for a case and 0 for a control), and also obtaining relevant covariates in the full cohort on each case and control for analysis of the resulting data by conditional logistic regression.

In the first step, we create '***matched_cases_controls***', each record consisting of a case with a matched control as well as relevant covariates of interest and the follow-up time *tcens* (extracted from '***fullset***'). In the second, we make the case_id (i.e. *pateid_case*) the identifier (i.e. *pateid*) and then create in the third step, the dataset '***id***' where we generate an identifier for each stratum (i.e. '*matchset*') to link the cases with their corresponding matched sets. Next we sort '***id***' by the control's id (*pateid*) to create '***id_case***'. Then in the fourth step, we match the two datasets '***matched_cases_controls***' and '***id_case***' by pateid (i.e. the control id) and obtain follow-up time and if needed, generate updates of covariates for each matched control. This can be a critical step: for time-fixed covariates not involving in the matching (i.e. ethnicity, social class), this can be obtained at index date. However, any that are time-dependent (for example, cumulative exposure level, duration of an important comorbidity), updated values of the covariates will need to be obtained on both case and its matched controls at the event time (*tcens*). In other words, such covariates can only be calculated after construction of the risk sets and as at the time the case experienced the event. Indeed, the final two steps are used to complete construction of the individual risk sets (i.e. 2×2 blocks), thus completing the data preparatory steps for analysis by conditional logistic regression.

Conclusions

Observational studies involving the nested case–control design are now commonplace and the trend of increased usage of the design in preference over the much simpler cohort design is likely to accelerate over time for a number of reasons, already described in epidemiological literature.[8–11] The problem of time-varying treatment arguably provides the main rationale for the popularity of the nested case–control design in safety studies and today, most published pharmacoepidemiological safety studies are based on the design despite the known problems of bias associated with the design.[11,12] To minimize the potential effects of confounding factors — the main source of bias — we match cases to controls on at least, the most important confounding covariates in preference over modeling their effects. This is a process that involves sampling of potential controls and although there are many sampling approaches, that of incidence density sampling is widely recognized as the best. From programming perspective, the main challenges depend on the size of the control population and appropriate implementation of the recommended sampling approach.

In this paper, we have described a simple approach for an accurate implementation of the nested case–control design based on a SAS program, generic enough for easy adaption to any given study involving the design. Our approach may be first to utilize the matching provision offered by SAS PROC SQL to achieve exact incidence density sampling from the dynamic risk sets of the nested case–control design, while also ensuring appropriate implementation of all the vital conditions of the design.

Acknowledgements

For their encouragement and material support, the author would like to thank Knut Muller and Sasha Ahrweiler of UCB Biosciences GmbH as well as to dedicate this work to the memory of my former research colleague Dr George Visick.

References

1 Vandenbroucke JP. What is the best evidence for determining harms of medical treatment? CMAJ. 2006;**174**(5):645–6.
2 Lubin JH, Gail MH. Biased selection of controls for case–control analyses of cohort studies. Biometrics. 1984;**40**:63–75.
3 Robins JM, Gail MH, Lubin JH. More on 'Biased selection of controls for case–control analyses of cohort studies'. Biometrics. 1986;**42**:293–9.
4 Langholz B, Richardson DB. Fitting general relative risk models for survival time and matched case–control analysis. Am J Epidemiol. 2010;**171**:377–83.
5 Richardson DB. An incidence density sampling program for nested case–control analyses. Occup Environ Med. 2004;**61**(12):e59.
6 Beaumont JJ, Steenland K, Minton A, Meyer S. A computer program for incidence density sampling of controls in case–control studies nested within occupational cohort studies. Am J Epidemiol. 1989;**129**:212–9.
7 Pearce N. Incidence density matching with a simple SAS computer program. Int J Epidemiol. 1989;**18**:981–4.
8 Stampfer MJ. ITT for observational data: worst of both worlds? Epidemiology. 1995;**6**:248–53.
9 Miettenen OS, Caro JJ. Principles of nonexperimental assessment of excess risk, with special reference to adverse drug reactions. J Clin Epidemiol. 1989;**42**(4):325–31.
10 Guess HA. Behavior of the exposure odds ratio in a case–control study when the hazard function is not constant over time. J Clin Epidemiol. 1989;**42**(12):1179–84.
11 Kiri VA, McKenzie G. How real is intention-to-treat (ITT) analysis in non-interventional PASS? We can do better. Curr Drug Saf. 2009;**4**(2):137–42.
12 Kiri VA. A pathway to improved prospective observational post-authorization safety studies. Drug Saf. 2012;**35**(9):711–24.