In the process of recording information and data entry, variation in identifiers information used to discover clusters of records pertaining to the same individual can arise through confusion of similar sounding names (such as Wang Fang and Wang Fang), names that appear visually similar (e.g. \_\_), keystroke errors, or the use of alternative names. Such distortions can bring difficulties to the subsequent data processing and data analysis process. For example, in the analysis of hepatitis B reported cases, since hepatitis B cannot be cured at present, in the process of calculating the incidence rate, the previously reported cases should be excluded, but if the patient name of the report is not completely consistent with the previous records, using the exact match, it is impossible to accurately identify the previously reported records. Thus fuzzy matching or probabilistic methods, which account for the probability that two sets of information refer to the same individual in the presence of errors, are important for information retrieval.

When identifying information is available in multiple fields, we can apply probabilistic record linkage methods to estimate the probability that pairs of records match given given patterns of agreement between their fields (e.g. same name, same date of birth, same sex, different address). This process can be improved by fuzzy matching within fields as well, which is the purpose of name matching algorithms which predict whether two names can refer to the same individual (e.g. \_\_ and \_\_ are somewhat likely to be a match, while \_\_ and \_\_ are not).

The existing name matching algorithms are mostly based on English text development and are not applicable to Chinese. For the Chinese name, the first name mostly includes only 2-4 Chinese characters, and unlike English, the format under which Chinese characters are stored in computer memory does not reveal their pronunciation. If you like the English name, only the [editing distance similarity](https://www.cnblogs.com/ivanyb/archive/2011/11/25/2263356.html) between characters will be considered , which will bring great deviation to the result. For example, from Cheng Jinjin to Cheng Yijin, the edit distance is 1, the edit distance similarity is 0.66. Thus, although it’s plausible that a data entry clerk could mishear Cheng Jinjin as Cheng Yijin, it will be difficult for the default edit distance algorithm to distinguish this potential match from a large number of nonmatches (e.g. from the Chinese habit of giving siblings names that differ by only one character, such as Jiang Jingguo and Jiang Weiguo, Song Qingling and Song Meiling). In contrast, the edit distance from Charles to Chales is 1, and the edit distance similarity is 0.86. Therefore, it is not enough to match Chinese names by editing distance similarity. Furthermore, there are a large number of Chinese characters, many of which differ by a small number of radicals or strokes, and so visually similar characters can be confused during data entry (e.g. \_\_ ). Thus we need some way of extracting phonetic and visual characteristics of Chinese names, as well as the potential sequence of keystrokes used to enter them in order to get good name matching performance. We extract these characteristics via a set of transformations including converting the names to pinyin, Wubi86, four corner code, and decomposing them into their component radicals and structural encodings. After transforming the names, we estimate one or more measures of similarity between name pairs, such as edit similarity, cosine similarity (less sensitive to reordering of characters), and longest common substring similarity (less sensitive to the presence of extra characters), and optionally combine them via trained classification models to predict the match status.

Based on the characteristics of Chinese characters and Chinese names, we have improved an existing R package for efficient probabilistic record linkage, fastLink, and incorporated name matching methods ranging from the use of a single transformation / similarity measure, a linear combination of several transformation / similarity measure pairs, and boosted decision tree models using the machine learning algorithm xgboost into fastLink. The name matching methods can simultaneously consider the pronunciation of Chinese characters (Pinyin), glyphs (five pens and four corner numbers), radicals and structures. In addition to the name, other attributes such as gender, date of birth, address, etc. can be considered in the record matching. This document describes how to install the enhanced fastLink package, how to use it, and how to interpret the results.

**2. Installation**

1. [Install R and RStudio](https://blog.csdn.net/Joshua_HIT/article/details/73741139) . See the [R language primer](https://zhuanlan.zhihu.com/p/45503712) for the use of Rstudio .
2. [Install Rtools](https://www.cnblogs.com/liugh/p/9937489.html) - only steps one and two are required, installing and setting environment variables
3. [Install the required R packages](http://blog.sciencenet.cn/blog-2379401-936653.html) tidyverse, xgboost and devtools, which can be installed according to Method 1 in the link. Enter install.packages("tidyverse") in the RStudio console and press Enter. The other two packages can be installed in the same way.
4. Use the following code to install the package chinsimi, fastLink developed by Berkeley.

devtools::install\_github('OPTI-SURVEIL/fastLink',dependencies = T, force = TRUE)

devtools::install\_github('OPTI-SURVEIL/chinsimi',dependencies = T, force = TRUE)

**3. Preparation**

1. Download all the files in [this folder](https://github.com/OPTI-SURVEIL/RLManual) except README.md, where:

* Name match 1.csv and Name match 2.csv as sample data
* linkage\_utils.R contains several functions used in the matching process
* filled F-curves.Rdata contains information used to determine the optimal threshold score at which names should be considered a match during the linkage process.

1. Set the system environment to Chinese. Because the records we need to match are in Chinese, we need to first set the R environment to Chinese. On windows machines, the command is as follows.

Sys.setlocale(category = 'LC\_ALL', locale = 'Chinese')

1. Set the working folder to save the folder where the data is downloaded in step 1. Note that the file path should use "/" instead of "". For example, "C:\Users\Documents" should not be used instead of "C:/Users/". Documents/". E.g:

setwd("C:/Users/Documents/")

**Replace C:/Users/Documents/ with the path to save the downloaded data in step 1.**

1. Install and load required R packages

pkgs = c(‘tidyverse’,’fastLink’,’xgboost’) #list of packages to install

toinstall = !(pkgs %in% installed.packages()) #check which ones are not already installed

lapply(pkgs[toinstall],install.packages) #install packages not already installed

library(tidyverse) #load packages into R

library(fastLink)

library(ChinSimi)

library(xgboost)

1. Import the R file linkage\_utils.R that includes the required function (downloaded in step 1)

source('linkage\_utils.R')

1. Import name matching machine learning model and threshold data

load('final\_xgb\_model\_10.Rdata')

load('filled F-curves.Rdata')

**4. How to use**

1. Import Data

fastLink bag with Fellegi-Sunter record matching process, see the detailed description [of this article](https://imai.fas.harvard.edu/research/files/linkage.pdf) . The main functions in the package are fastLink() and getMatches(). fastLink() is used for matching, and getMatches() is used to extract matching records. The data is read first. The following is the reading example data, where S1 is the data 1 that needs to be matched, and S2 is the data 2 that needs to be matched. The following is the sample data downloaded before use. The correct matching result is that the 1-100 lines in S1 correspond one-to-one with the 1-100 lines in S2. **Note: you** need to set stringAsFactors = FALSE, otherwise the name will be read as a factor instead of a string, and the subsequent matching process will report an error.

S1 <- read.csv("Name match 1.csv", stringsAsFactors = FALSE)

S2 <- read.csv("Name match 2.csv", stringsAsFactors = FALSE)

You can see which data is included in S1 and S2 by the following commands.

View(S1)

View(S2)

The following are the first 6 lines of S1:

name sex yob mob dob

1 孙文 0 1975 6 1

2 莊子 1 1980 10 17

3 伊姆荷太普 1 1993 4 6

4 神农氏 1 1983 11 9

5 陈水扁 0 1977 9 6

6 拿破仑 1 1958 8 19

The following are the first 6 lines of S2:

name sex yob mob dob

1 孫中山 0 1975 6 1

2 庄子 1 1980 10 17

3 印何闐 1 1993 4 6

4 神农 1 1983 11 9

5 陳水扁 0 1977 9 6

6 拿破仑一世 1 1958 8 19

The name field is the name, the sex field is the gender (randomly generated, does not represent the true gender), yob is the year of birth, mob is the month of birth, and dob is the date of birth (randomly generated, does not represent the real date of birth). NA indicates that the field data is missing. **NOTE:** These data were used in a previous paper about Chinese name matching on Wikipedia links, and are provided for convenience. They contain several types of name variation that were not used during our methods development (e.g. simplified vs. traditional Chinese, pseudonyms, and honorifics), and so name matching performance on this dataset may not reflect the intended application of our methods.

1. Data pre-processing

Often, we may first wish to clean the data we are planning to link. This step is not necessary for the example datasets provided, but we illustrate it briefly below for another set of names where we have introduced commonly observed types of noise.

First we read in the data

messynames = read\_csv('messy\_names.csv')

Then check for alphabetic characters (using the function grep) and remove them where appropriate (using the function gsub):

alphainds = grep('[a-z]',messynames$name, ignore.case = T) #pattern to find, where to look for it, ignore.case will find both lowercase and uppercase lettersmessynames$name[alphainds]

[1] "夏禹L" "宋慶齡D" "穆圣S" "陶吉吉I" "勾踐G" "韩幌O" "宣統帝X" "呂后J" "小萝卜头I"

[10] "波布F" "唐泽寿明B" "梅澤由香里J" "伍豪Q" "順治帝I" "赖传湘V" "方臘H" "费祎T" "苏我马子T"

[19] "近衛文麻呂Q" "衛子夫A." "管谟业Y" "孙武子L" "底比斯圣团J" "源義経E" "松川尚琉辉T" "唐紹儀Q" "吳志昊T"

[28] "佐橋俊彦I" "蘇蕙L" "張愛萍W" "蘇文忠N" "古尔班M" "仲達Q" "爆料天王K" "刻?俄斯B" "吳惠蘭O"

[37] "史太君5X" "阿尔戈英雄V" "羅一民R" "胡传A" "范長生M" "蔡时那R" "许悼公Q" "许素叶B" "譚家述S"

[46] "赖幸媛V" "赵观涛N" "邱君强S" "邹普胜T" "金溥聪L" "关口知宏R" "冯素弗H" "高桥冴未V" "高桥绍运C"

[55] "黄平洋U" "黃乃裳0A" "黃苑玲F" "趙宣子K" "呂柏克Z" "藤井リナV" "张仲景Z" "朴志恩B" "子龍B"

[64] "朴正洙D" "彭和尚Y" "瑶姫P" "乃颜之乱F" "水嶋寬A" "高河ゆんE" "玉琳国师N" "王会龙J" "鬼虎之亂L"

[73] "宣尼公M" "林巧雅G" "池内莉奈V" "藤原頼忠P" "季孙强H" "公子寬K" "薛氏利忠A" "耶律喜孙Y" "韩德昌S"

[82] "山内鈴蘭Z" "楚王子陽W" "斉藤由貴I" "白磊R" "葵ちひろY" "精济H" "赫波N" "梅泽由香里T" "王女自鳴鼓V"

[91] "宋渔父C" "武德尊侯C" "关羽云长U" "平措旺阶T" "王龍舒N" "张宗可G" "脱火赤N" "荀罌O"

It appears that we can safely remove all the alphabetic characters, as none of these names should have them:

messynames$name[alphainds] = gsub('[a-z]','',messynames$name[alphainds],ignore.case = T) #pattern to replace, what to replace it with,where to replace it, ignore.case will replace both lowercase and uppercase letters

messynames$name[alphainds]

[1] "夏禹" "宋慶齡" "穆圣" "陶吉吉" "勾踐" "韩幌" "宣統帝" "呂后" "小萝卜头" "波布"

[11] "唐泽寿明" "梅澤由香里" "伍豪" "順治帝" "赖传湘" "方臘" "费祎" "苏我马子" "近衛文麻呂" "衛子夫."

[21] "管谟业" "孙武子" "底比斯圣团" "源義経" "松川尚琉辉" "唐紹儀" "吳志昊" "佐橋俊彦" "蘇蕙" "張愛萍"

[31] "蘇文忠" "古尔班" "仲達" "爆料天王" "刻?俄斯" "吳惠蘭" "史太君5" "阿尔戈英雄" "羅一民" "胡传"

[41] "范長生" "蔡时那" "许悼公" "许素叶" "譚家述" "赖幸媛" "赵观涛" "邱君强" "邹普胜" "金溥聪"

[51] "关口知宏" "冯素弗" "高桥冴未" "高桥绍运" "黄平洋" "黃乃裳0" "黃苑玲" "趙宣子" "呂柏克" "藤井リナ"

[61] "张仲景" "朴志恩" "子龍" "朴正洙" "彭和尚" "瑶姫" "乃颜之乱" "水嶋寬" "高河ゆん" "玉琳国师"

[71] "王会龙" "鬼虎之亂" "宣尼公" "林巧雅" "池内莉奈" "藤原頼忠" "季孙强" "公子寬" "薛氏利忠" "耶律喜孙"

[81] "韩德昌" "山内鈴蘭" "楚王子陽" "斉藤由貴" "白磊" "葵ちひろ" "精济" "赫波" "梅泽由香里" "王女自鳴鼓"

[91] "宋渔父" "武德尊侯" "关羽云长" "平措旺阶" "王龍舒" "张宗可" "脱火赤" "荀罌"

Now we do the same thing for numeric characters:

numberinds = grep('[0-9]',messynames$name)

messynames$name[numberinds]

[1] "阿倍仲麻吕9" "子思1" "公输般8" "湖南少年歌2" "奧爾甫斯9" "后梁郢王2" "陈果夫5" "矢泽爱3"

[9] "冈崎律子9" "鄭夢九1" "刘镇伟7" "胡适之7" "曹无伤8" "辜显荣9" "又吉耶稣9" "悠仁1"

[17] "陈希周9" "阿?坦3" "完顏兀朮7" "坤興公主4" "郭紹培0" "嶽帝3" "上門導師5" "韩佳人8"

[25] "吕太后8" "晉文帝3" "關山月7" "鄧先聖2" "張經武7" "吴景9" "天勇星1" "纪成4"

[33] "李少荃3" "矢岛保治郎7" "秦钟8" "窦泰5" "史太君5" "胡综7" "胡义宾2" "庄和子2"

[41] "董志宁7" "蔡明晋6" "蔡昱详2" "萧思江7" "苏过4" "袁涣3" "郭图4" "郑浑4"

[49] "郦琼1" "金胜友8" "金聖愛0" "金泳镇3" "关勉3" "韦奇立3" "顧順章9" "馬敘倫7"

[57] "高桥(美)由纪4" "黃乃裳0" "武藤啓9" "聂赤赞布2" "安蒂岡妮7" "杨镇龙4" "南漢烈宗8" "儀間真常7"

[65] "神格化0" "明宣帝8" "朴政玟5" "巴布爾6" "湖月わたる5" "僖叔1" "张师正3" "岩本えり子9"

[73] "城崎麻理子2" "桜井菜々子4" "徐官喜4" "中行偃1" "神話織女0" "楊洪勝5" "郭定文4" "白鴻(亮)3"

[81] "段芝泉7" "愛音まひろ1" "鲍宣1" "毛利慶親9" "韓隤當8" "上橋菜穂子7" "平陽君2" "隱元隆琦2"

[89] "独孤颎3" "圆馆金2" "琼波奈觉5" "宝亲王1" "宗室博和託6" "水戸光子5" "夏蓮居7" "黄琬5"

[97] "郭華倫2" "徐允恭2" "莽骨速3" "高二哥3"

Again, it doesn’t appear that there’s any reason to keep these numbers, so we can remove them all:

messynames$name[numberinds] = gsub('[0-9]','',messynames$name[numberinds])

And finally, we examine entries with punctuation:

punctinds = grep('[[:punct:]]',messynames$name)

messynames$name[punctinds]

[1] "村上春树." "王?維" "周(恩)來" "華?鋒" "夏侯淵}" "伊姆霍特普," "孙立人." "光緒帝?"

[9] "楊業?" "吉娃斯阿麗?" "谢仁祖." "高健龙." "華陀?" "林延年." "陳友諒," "朴赞郁."

[17] "惠(美)押勝" "汪精衛," "秦二世胡亥." "刘旻?" "中岛(美)雪" "劉安?" "天上聖母," "藤本(美)贵"

[25] "锦户(亮)" "三浦(美)紀" "李泽楷." "吳?璋" "鄧世昌}" "稲垣?郎" "中谷(美)纪" "伊东(美)咲"

[33] "泽城(美)雪" "余登發?案" "金万重," "海王(美)智留" "羅賓漢}" "明石守重," "道姑?" "高橋留(美)子"

[41] "諸葛瑾}" "冈村明(美)" "志田未来}" "張?燾" "陳?峻" "邹至蕙}" "许筠?" "軒轅."

[49] "小室圭子." "保羅遮打?" "衛子夫." "紺野麻(美)" "安達充." "阿?坦" "徳永千奈(美)" "筱原惠(美)"

[57] "虞妙弋." "荒川弘(美)" "顏惠慶}" "徐羡之?" "德川光?" "陶絮忠?" "副香港總督," "伊東(美)笑"

[65] "鄭?姓" "竹岡(美)穂" "?民哀悼日" "?殤日" "詹姆斯.溫" "中岛(美)嘉" "蘇子(美)" "杜子(美)"

[73] "王中孚." "王雲龍}" "(美)水かがみ" "溫王(爺)" "李德鄰?" "金鈺彬," "刻?俄斯" "ひと(美)"

[81] "夏?璋" "萬善(爺)" "宍戸留(美)" "文?" "丁濟(美)" "三?人" "相贺照?" "三條實(美)"

[89] "關聖(恩)主" "下村务}" "福圆(美)里" "秦?舫" "童琮辉?" "竹冈(美)穗" "中村爱(美)" "長狄?"

[97] "黒河奈(美)" "陈硕贞," "紹爾兄(妹)" "罗志(恩)" "罗(美)洛" "(美)水镜" "(美)浓部达吉" "(美)空云雀"

[105] "(美)竹凉子" "兴梠里(美)" "华原朋(美)" "薩?剌" "萧(美)琴" "蓝成春," "蓝(美)津" "渡辺久(美)子"

[113] "西?从道" "许纯(美)" "谢(美)惠" "詩琳通公主}" "丰原里(美)" "赫尔南?兹" "趙振東," "趙貴,"

[121] "赵可怀?" "赵家(骧)" "邵传勇}" "郗鑒?" "柳瀨夏(美)" "钱思(亮)" "关劲松." "閻迦?"

[129] "陳?砥" "陈婉真." "陈峰民?" "陈希(亮)" "陈永?" "陈江和}" "陈(美)玉" "陈铭枢}"

[137] "李俊基." "陆文声." "韦(骧)" "顾崇廉," "顧仁(恩)" "飞田展男}" "饒?華" "蔣宋(美)齡"

[145] "高桥久(美)子" "高桥广树}" "高桥(美)由纪" "鱼朝(恩)" "鹤弘(美)" "黄子澄}" "黄宏发," "铃木亚(美)"

[153] "下村修}" "一条(美)賀子" "辣(妹)曾根" "中山(美)穂" "周淑(美)" "久保田和輝." "林秀榮?" "中岛裕(美)子"

[161] "別府歩(美)" "四大貝?" "阿?坦汗" "多?之父" "寶嘉康蒂," "郭洪福," "宋?鋒" "林白水,"

[169] "浩(亮)" "陳秀(美)" "豊原里(美)" "索爾兄(妹)" "米沢瑠(美)" "阿布賴." "?頭正彌" "?頭左馬守"

[177] "?頭盛順" "池府王(爺)" "大亀明日香," "韓(恩)廷" "胡润百富." "汤槱}" "護?尊王" "攸(美)尼斯"

[185] "星(美)里" "汉昭烈帝." "柳在?" "高天?" "愍周皇后?" "新垣仁絵}" "日葉洲媛?" "碇真治,"

[193] "真由(美)" "台灣?王" "安朵(美)達" "槻潮钢}" "田中(美)久" "菜菜(美)ねい" "岩下(美)季" "当山ひとみ?"

[201] "绍尔兄(妹)" "朔爾兄(妹)" "索尔兄(妹)" "安室奈(美)恵" "边荣立}" "孫(美)子" "壽(耆)" "(終)南捷徑"

[209] "磐井?" "川島省?" "荀躒," "中??父" "子 (姓)" "白鴻(亮)" "一條愛(美)" "大沢(美)加"

[217] "希(美)まゆ" "程定國}" "明智珠," "青野真衣," "大眾(爺)" "艾末末}" "林朝璣}" "孙文子?"

[225] "山本杏(美)" "違?侯" "江南?主" "崔.丹尼爾" "?尔锦" "?靈的剋星" "朝比奈ハル}" "迷男,"

[233] "彥五瀨?" "伊(耆)放勳" "朝樹りさ}" "德色赉?布" "宮部(美)雪" "陈尧初." "靏弘(美)" "李債潾."

[241] "鶴岡聡}" "高宣子." "雄诺敦?布" "松本あゆ(美)" "幾瑟," "文(耆)" "枡田絵理奈?" "猿田彥?"

[249] "王信?" "黒瀬真奈(美)" "薛超?" "寛仁親王," "吹石一恵." "?古尔苏荣" "杨次山?" "瀬戸麻沙(美)"

[257] "明坂聡(美)" "浜崎亞由(美)" "濱崎亞由(美)" "者?篾" "宇佐(美)吉啓" "?軍日" "井上雅(美)" "李繼(儼)"

[265] "桃栗蜜?" "謝幼安}" "豁埃马阑?"

This case is more complicated. Some characters, e.g. **. , }** seem like they can be removed without losing any information. However other characters, such as **()**  and **?** may indicate that a character was present but the data entry personnel couldn’t identify it. These markers of ambiguity are taken into account by our name matching methods, and so for now we recommend leaving them in. Thus, we remove only a subset of the punctuation characters.

messynames$name[punctinds] = gsub('.','',messynames$name[punctinds],fixed = T) #fixed replaces exactly the pattern provided, without performaing any regular expression matching

messynames$name[punctinds] = gsub(',','',messynames$name[punctinds],fixed = T)

messynames$name[punctinds] = gsub('}','',messynames$name[punctinds],fixed = T)

Finally, we remove any whitespace

messynames$name = gsub('[[:space:]]','',messynames$name)

1. Record match (simple version)

Record matching using the fastLink() function. In the R console, enter ?fastLink to view the usage instructions for the function.

valres <- fastLink(dfA = S1, dfB = S2,

varnames = c('name','sex','yob','mob','dob'),

stringdist.match = 'name',

stringdist.method = chin\_strsim,

stringdist.args = list(model = model\_10, reftable = unique(S1$name, S2$name)),

string.transform = transparser,

string.transform.args = list(model = model\_10,reftable = unique(S1$name, S2$name)),

cut.a = xgb10F1\_filled$opt.thresh,

verbose = T, estimate.only = F, cond.indep = F)

* dfA represents the first record set that needs to be matched, and dfB represents the second record set that needs to be matched. Here, since we are naming the data set named S1 and S2 when reading the data, set dfA = S1 when inputting. dfB = S2. In practice, you need to set it according to the name of the dataset you used when reading the data. It should be noted that in S1 and S2, please use a consistent field name, and it is best to use the English field name. For example, if the name field is named xingming in S1, then it should be consistent in S2, but not name. In order to perform internal linkage or deduplication, provide the same dataframe to both dfA and dfB.
* Varnames represents the fields that need to be used for matching. In S1 and S2, we match all five fields, name, sex, yob, mob, and dob.
* Stringdist.match lists the fields that we want to apply a string based fuzzy match to. The example here indicates that we need to use our Chinese matching method to match the fields, that is, through pinyin, four-corner number, five strokes, radicals, font structure and their combination to match. Here we only use Chinese for the name field. String matching. In practice, please set the name of the field you need to match your name. If no field names are provided to Stringdist.match, the algorithm will define agreement on each field by exact matching.
* Stringdist.method represents a function for calculating the similarity of a name whose return value is a matrix representing the similarity of each element in S1 and S2. The chin\_strsim function is used here. **Please do not modify this parameter unless you know what you are doing.**
* Stringdist.args means to enter the parameters of the chin\_strsim function or another custom function provided to Stringdist.method. The arguments provided to chin\_strsim include either a variable name or model object, which will be parsed to see which transformations and similarity measures need to be calculated, and how to combine multiple similarity measures via the provided model. Additionally, since some of our methods consider the frequency of surnames and given names (since a match on a common surname is less informative than a match on a rare surname), a list of names from which to calculate frequencies may be needed. Replace S1 and S2 in reftable = unique(S1$name, S2$name) with the names of dfA and dfB you use, and replace the name with the one you are using.
* String.transform represents a function used to generate pinyin, four-corner numbers, five strokes, radicals, font structures, and combinations thereof. The transparser function is used here. **Please do not modify this parameter unless you know what you are doing.**
* String.transform.args represents the parameters input to the transformation function. transparser accepts the same arguments as chin\_strsim, except for some additional options not mentioned here. Same as above, for the name frequency table, replace S1 and S2 in reftable = unique(S1$name, S2$name) with the names of dfA and dfB you use, and replace the name with you.
* Cut.a indicates the threshold of similarity in judging whether the names fully match. Our current approach calculates this threshold based on validation data used to develop the name matching models. The threshold may be optimally adjusted for each linkage application, which will be discussed in section 4.
* Verbose determines the degree of detail displayed for matching progress. When T is set to indicate yes, it is set to F to indicate no. We recommend to leave this parameter set to T.
* Estimate.only indicates whether to output only the parameters, and does not output the matching result. When T is set, it means yes, that is, only the parameters of the model are output. If it is recommended to be F, the parameters and matching results can be output at the same time.
* Cond.indep indicates whether agreement between identifying fields is assumed to be independent (e.g. names are equally likely to randomly match whether or not address also matches). It is recommended to set to F.

During the matching process, the information similar to the following will be output.

====================

fastLink(): Fast Probabilistic Record Linkage

====================

If you set return.all to FALSE, you will not be able to calculate a confusion table as a summary statistic.

Calculating matches for each variable.

Matching variable name using string-distance matching.

WARNING: You have no exact matches for name.

Matching variable sex using exact matching.

Matching variable yob using exact matching.

Matching variable mob using exact matching.

Matching variable dob using exact matching.

Calculating matches for each variable took 0.7 minutes.

Getting counts for parameter estimation.

Parallelizing calculation using OpenMP. 1 threads out of 8 are used.

Getting counts for parameter estimation took 0 minutes.

Running the EM algorithm.

Running the EM algorithm took 0.29 seconds.

Selected match probability threshold is: 0.254680688264359

Getting the indices of estimated matches.

Parallelizing calculation using OpenMP. 1 threads out of 8 are used.

Getting the indices of estimated matches took 0 minutes.

Deduping the estimated matches.

Deduping the estimated matches took 0 minutes.

Getting the match patterns for each estimated match.

Getting the match patterns for each estimated match took 0 minutes.

1. Extract matching records

To extract matching records, you need to use the getMatches function. We will set the threshold linkage probability for matches to be the threshold selected by the algorithm, but in practice you may choose a different number.

matched\_dfs <- getMatches(dfA = S1, dfB = S2, fl.out = valres, threshold.match = valres$EM$threshold.match, combine.dfs = FALSE, twolineformat = TRUE)

* dfA and dfB represent two datasets that need to be linked. Again, for internal linkage, provide the same dataframe to both arguments.
* fl.out represents the output of fastLink, here valres
* threshold.match represents the cutoff matching probability for retrieving linked record pairs. Here we use the optimal threshold estimated during the linkage process.
* When combine.dfs is set to F or FALSE, two data tables are output. When T or TRUE is set, the data table in which the matching records are merged is output. This merged table can be useful in cases where e.g. and independent and dependent variable are contained in separate datasets.
* Twolineformat is an additional option for formatting the output, which may be convenient when reviewing the quality of the links. When set to T(RUE), the matching records are displayed together. Only works if combine.dfs is set to F(ALSE).

The following are sample results when combine.dfs, twolineformat is set to T or F.

* combine.dfs = T, twolineformat = F, the result is that the two recordsets are displayed together, showing only the individual properties in dataset A. Gamma.name, gamma.sex... respectively represent whether each attribute matches, 0 means no match, 2 means match. In the first line, gamma.name = 0, which means that the names are inconsistent in the first matching record.

name sex yob mob dob gamma.name gamma.sex gamma.yob gamma.mob gamma.dob

1 孙文 0 1975 6 1 0 2 2 2 2

2 莊子 1 1980 10 17 0 2 2 2 2

3 伊姆荷太普 1 1993 4 6 0 2 2 2 2

4 神农氏 1 1983 11 9 0 2 2 2 2

5 陈水扁 0 1977 9 6 2 2 2 2 2

6 拿破仑 1 1958 8 19 0 2 2 2 2

* combine.dfs = F, twolineformat = F At this point, the matching records are displayed as two data sets, that is, the first line in dfA.match corresponds to the first line in dfB.match.

$`dfA.match`

name sex yob mob dob gamma.name gamma.sex gamma.yob gamma.mob gamma.dob posterior

1 孙文 0 1975 6 1 0 2 2 2 2 0.9891703137473733

2 莊子 1 1980 10 17 0 2 2 2 2 0.9891703137473733

3 伊姆荷太普 1 1993 4 6 0 2 2 2 2 0.9891703137473733

4 神农氏 1 1983 11 9 0 2 2 2 2 0.9891703137473733

5 陈水扁 0 1977 9 6 2 2 2 2 2 0.9999983070577794

6 拿破仑 1 1958 8 19 0 2 2 2 2 0.9891703137473733

$dfB.match

name sex yob mob dob gamma.name gamma.sex gamma.yob gamma.mob gamma.dob posterior

1 孫中山 0 1975 6 1 0 2 2 2 2 0.9891703137473733

2 庄子 1 1980 10 17 0 2 2 2 2 0.9891703137473733

3 印何闐 1 1993 4 6 0 2 2 2 2 0.9891703137473733

4 神农 1 1983 11 9 0 2 2 2 2 0.9891703137473733

5 陳水扁 0 1977 9 6 2 2 2 2 2 0.9999983070577794

6 拿破仑一世 1 1958 8 19 0 2 2 2 2 0.9891703137473733

* combine.dfs = F, twolineformat = T now matches the matching records

row.index name sex yob mob dob p\_match

1 dfA.1 孙文 0 1975 6 1

2 dfB.1 孫中山 0 1975 6 1

3 agreement pattern: 0 2 2 2 2 0.9892

4

5 dfA.2 莊子 1 1980 10 17

6 dfB.2 庄子 1 1980 10 17

7 agreement pattern: 0 2 2 2 2 0.9892

8

9 dfA.3 伊姆荷太普 1 1993 4 6

10 dfB.3 印何闐 1 1993 4 6

11 agreement pattern: 0 2 2 2 2 0.9892

12

13 dfA.4 神农氏 1 1983 11 9

14 dfB.4 神农 1 1983 11 9

15 agreement pattern: 0 2 2 2 2 0.9892

16

17 dfA.5 陈水扁 0 1977 9 6

18 dfB.5 陳水扁 0 1977 9 6

19 agreement pattern: 2 2 2 2 2 1

20

21 dfA.6 拿破仑 1 1958 8 19

22 dfB.6 拿破仑一世 1 1958 8 19

23 agreement pattern: 0 2 2 2 2 0.9892

1. Refining the threshold for name matching

The optimal threshold for declaring matching name pairs depends on the expected ratio of matched record pairs to non-matches, as well as the quality of information provided by other fields used in linkage. If the expected ratio of matches is relatively high, and other available identifiers provide good discrimination between matching and non-matching records, we can set a lower threshold for name matching, increasing linkage sensitivity with little risk of retrieving an excess of false positive links.

In order to set an estimated optimal threshold for any linkage application, our suggested workflow is to first run a linkage based on exact matching to estimate the ratio of matching record pairs, as well as the informational content of the available identifiers.

exact\_match\_res <- fastLink(dfA = S1, dfB = S2,

varnames = c('name','sex','yob','mob','dob'),

verbose = T, estimate.only = F, cond.indep = F)

====================

fastLink(): Fast Probabilistic Record Linkage

====================

If you set return.all to FALSE, you will not be able to calculate a confusion table as a summary statistic.

Calculating matches for each variable.

Matching variable name using exact matching.

WARNING: You have no exact matches for name. Matching variable sex using exact matching.

Matching variable yob using exact matching.

Matching variable mob using exact matching.

Matching variable dob using exact matching.

Calculating matches for each variable took 0.43 minutes.

Getting counts for parameter estimation.

Parallelizing calculation using OpenMP. 1 threads out of 4 are used.

Getting counts for parameter estimation took 0.01 minutes.

Running the EM algorithm.

Running the EM algorithm took 1.37 seconds.

Selected match probability threshold is: 0.3841493573918854

Getting the indices of estimated matches.

Parallelizing calculation using OpenMP. 1 threads out of 4 are used.

Getting the indices of estimated matches took 0 minutes.

Deduping the estimated matches.

Deduping the estimated matches took 0.01 minutes.

Getting the match patterns for each estimated match.

Getting the match patterns for each estimated match took 0 minutes.

Now we can use the function F\_adjust\_link to estimate an optimal threshold for name matching. Arguments are:

* 1. Fcurve – an object storing the F1 score across multiple thresholds based on model validation data
  2. flinkres – A fastLink EM object
  3. thresh.match – the threshold probability for declaring matches from the fastLink run
  4. namecol – the name of the column in which names are stored
  5. plot – an option to plot the original (red) and calibrated (blue) F curves

adjusted\_F <- F\_adjust\_link(Fcurve = xgb10F1\_filled$curvedat,

flinkres = exact\_match\_res$EM,

thresh.match = exact\_match\_res$EM$threshold.match,

namecol = ‘name’,

plot = T)

Now we can run the fastLink algorithm with the optimized name matching criterion for this particular application

valres <- fastLink(dfA = S1, dfB = S2,

varnames = c('name','sex','yob','mob','dob'),

stringdist.match = 'name',

stringdist.method = chin\_strsim,

stringdist.args = list(model = model\_10, reftable = unique(S1$name, S2$name)),

string.transform = transparser,

string.transform.args = list(model = model\_10,reftable = unique(S1$name, S2$name)),

cut.a = adjusted\_F$opt.thresh,

verbose = T, estimate.only = F, cond.indep = F)

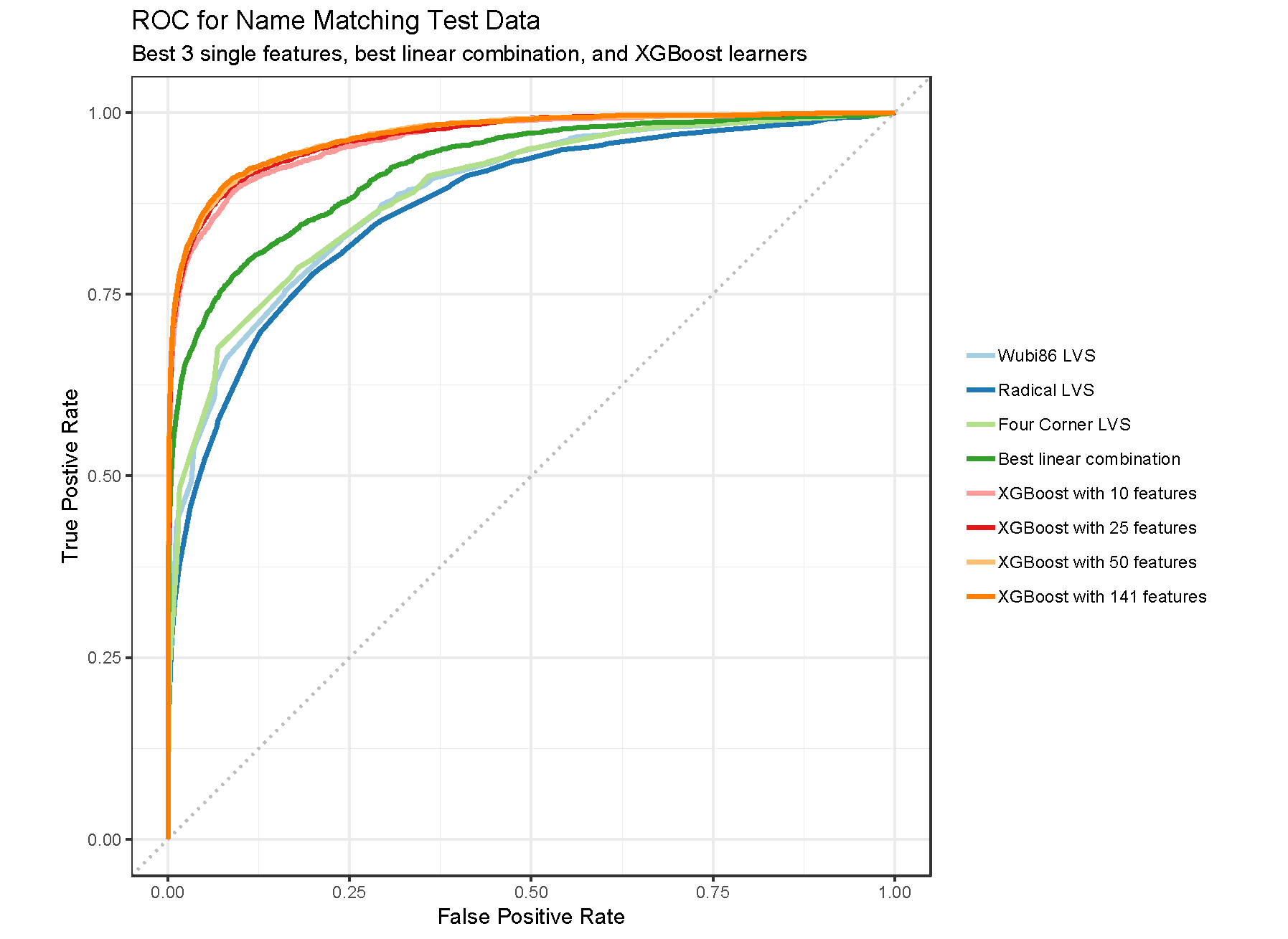
getMatches(S1, S2, valres, valres$EM$threshold.match, combine.dfs = F, twolineformat = T)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 146 | dfB.38 | 太公望 | 1 | 1978 | 11 | 25 |  |
| 147 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 148 |  |  |  |  |  |  |  |
| 149 | dfA.39 | 乾隆皇帝 | 1 | 1987 | 9 | 10 |  |
| 150 | dfB.39 | 乾隆帝 | 1 | 1987 | 9 | 10 |  |
| 151 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 152 |  |  |  |  |  |  |  |
| 153 | dfA.40 | 武則天 | 1 | 1995 | 6 | 24 |  |
| 154 | dfB.40 | 武则天 | 1 | 1995 | 6 | 24 |  |
| 155 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 156 |  |  |  |  |  |  |  |
| 157 | dfA.41 | 赵胜 | 0 | 1981 | 4 | 9 |  |
| 158 | dfB.41 | 平原君 | 0 | 1981 | 4 | 9 |  |
| 159 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 160 |  |  |  |  |  |  |  |
| 161 | dfA.42 | 耶穌基督 | 1 | 1981 | 12 | 25 |  |
| 162 | dfB.42 | 耶稣 | 1 | 1981 | 12 | 25 |  |
| 163 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 164 |  |  |  |  |  |  |  |
| 165 | dfA.43 | 唐尧 | 1 | 1983 | 1 | 20 |  |
| 166 | dfB.43 | 尧 | 1 | 1983 | 1 | 20 |  |
| 167 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 168 |  |  |  |  |  |  |  |
| 169 | dfA.44 | 毛澤東 | 0 | 1970 | 7 | 20 |  |
| 170 | dfB.44 | 毛泽东 | 0 | 1970 | 7 | 20 |  |
| 171 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 172 |  |  |  |  |  |  |  |
| 173 | dfA.45 | 馬殷 | 0 | 1979 | 3 | 2 |  |
| 174 | dfB.45 | 马殷 | 0 | 1979 | 3 | 2 |  |
| 175 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 176 |  |  |  |  |  |  |  |
| 177 | dfA.46 | 秦邦憲 | 0 | 1985 | 10 | 29 |  |
| 178 | dfB.46 | 博古 | 0 | 1985 | 10 | 29 |  |
| 179 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 180 |  |  |  |  |  |  |  |
| 181 | dfA.47 | 張作霖 | 1 | 1997 | 6 | 28 |  |
| 182 | dfB.47 | 张作霖 | 1 | 1997 | 6 | 28 |  |
| 183 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 184 |  |  |  |  |  |  |  |
| 185 | dfA.48 | 蒋经国 | 1 | 1972 | 11 | 16 |  |
| 186 | dfB.48 | 蔣經國 | 1 | 1972 | 11 | 16 |  |
| 187 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 188 |  |  |  |  |  |  |  |
| 189 | dfA.49 | 王國維 | 1 | 1993 | 11 | 4 |  |
| 190 | dfB.49 | 王国维 | 1 | 1993 | 11 | 4 |  |
| 191 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 192 |  |  |  |  |  |  |  |
| 193 | dfA.50 | 卢泰愚 | 1 | 1963 | 8 | 20 |  |
| 194 | dfB.50 | 盧泰愚 | 1 | 1963 | 8 | 20 |  |
| 195 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 196 |  |  |  |  |  |  |  |
| 197 | dfA.51 | 全斗焕 | 0 | 1994 | 6 | 24 |  |
| 198 | dfB.51 | 全斗煥 | 0 | 1994 | 6 | 24 |  |
| 199 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 200 |  |  |  |  |  |  |  |
| 201 | dfA.52 | 马英九 | 0 | 1975 | 4 | 18 |  |
| 202 | dfB.52 | 馬英九 | 0 | 1975 | 4 | 18 |  |
| 203 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 204 |  |  |  |  |  |  |  |
| 205 | dfA.53 | 朱厚熜 | 1 | 1973 | 10 | 19 |  |
| 206 | dfB.53 | 明世宗 | 1 | 1973 | 10 | 19 |  |
| 207 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 208 |  |  |  |  |  |  |  |
| 209 | dfA.54 | 甲骨学四堂 | 1 | 2002 | 9 | 31 |  |
| 210 | dfB.54 | 甲骨四堂 | 1 | 2002 | 9 | 31 |  |
| 211 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 212 |  |  |  |  |  |  |  |
| 213 | dfA.55 | 鄧小平 | 0 | 1990 | 2 | 10 |  |
| 214 | dfB.55 | 邓小平 | 0 | 1990 | 2 | 10 |  |
| 215 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 216 |  |  |  |  |  |  |  |
| 217 | dfA.56 | 齐天大圣 | 0 | 1996 | 9 | 21 |  |
| 218 | dfB.56 | 孙悟空 | 0 | 1996 | 9 | 21 |  |
| 219 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 220 |  |  |  |  |  |  |  |
| 221 | dfA.57 | 燧人 | 1 | 1958 | 8 | 14 |  |
| 222 | dfB.57 | 燧人氏 | 1 | 1958 | 8 | 14 |  |
| 223 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 224 |  |  |  |  |  |  |  |
| 225 | dfA.58 | 手冢治虫 | 0 | 1982 | 4 | 30 |  |
| 226 | dfB.58 | 手塚治虫 | 0 | 1982 | 4 | 30 |  |
| 227 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 228 |  |  |  |  |  |  |  |
| 229 | dfA.59 | 吕秀莲 | 1 | 1965 | 8 | 22 |  |
| 230 | dfB.59 | 呂秀蓮 | 1 | 1965 | 8 | 22 |  |
| 231 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 232 |  |  |  |  |  |  |  |
| 233 | dfA.60 | 倉頡 | 1 | 1971 | 4 | 17 |  |
| 234 | dfB.60 | 仓颉 | 1 | 1971 | 4 | 17 |  |
| 235 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 236 |  |  |  |  |  |  |  |
| 237 | dfA.61 | 張學良 | 0 | 1979 | 5 | 7 |  |
| 238 | dfB.61 | 张学良 | 0 | 1979 | 5 | 7 |  |
| 239 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 240 |  |  |  |  |  |  |  |
| 241 | dfA.62 | 裕仁 | 1 | 1985 | 1 | 7 |  |
| 242 | dfB.62 | 昭和天皇 | 1 | 1985 | 1 | 7 |  |
| 243 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 244 |  |  |  |  |  |  |  |
| 245 | dfA.63 | 耶稣基督 | 1 | 1971 | 2 | 17 |  |
| 246 | dfB.63 | 耶稣 | 1 | 1971 | 2 | 17 |  |
| 247 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 248 |  |  |  |  |  |  |  |
| 249 | dfA.64 | 漢高祖 | 0 | 1976 | 3 | 2 |  |
| 250 | dfB.64 | 刘邦 | 0 | 1976 | 3 | 2 |  |
| 251 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 252 |  |  |  |  |  |  |  |
| 253 | dfA.65 | 金太祖 | 1 | 1985 | 8 | 31 |  |
| 254 | dfB.65 | 完颜阿骨打 | 1 | 1985 | 8 | 31 |  |
| 255 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 256 |  |  |  |  |  |  |  |
| 257 | dfA.66 | 五虎上将 | 0 | 1974 | 4 | 15 |  |
| 258 | dfB.66 | 五虎将 | 0 | 1974 | 4 | 15 |  |
| 259 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 260 |  |  |  |  |  |  |  |
| 261 | dfA.67 | 耶穌 | 0 | 1974 | 8 | 13 |  |
| 262 | dfB.67 | 耶稣 | 0 | 1974 | 8 | 13 |  |
| 263 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 264 |  |  |  |  |  |  |  |
| 265 | dfA.68 | 溫家寶 | 0 | 1995 | 2 | 23 |  |
| 266 | dfB.68 | 温家宝 | 0 | 1995 | 2 | 23 |  |
| 267 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 268 |  |  |  |  |  |  |  |
| 269 | dfA.69 | 周恩來 | 0 | 1980 | 1 | 20 |  |
| 270 | dfB.69 | 周恩来 | 0 | 1980 | 1 | 20 |  |
| 271 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 272 |  |  |  |  |  |  |  |
| 273 | dfA.70 | 胡亥 | 0 | 1981 | 2 | 20 |  |
| 274 | dfB.70 | 秦二世 | 0 | 1981 | 2 | 20 |  |
| 275 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 276 |  |  |  |  |  |  |  |
| 277 | dfA.71 | 汉高祖 | 1 | 1993 | 11 | 29 |  |
| 278 | dfB.71 | 刘邦 | 1 | 1993 | 11 | 29 |  |
| 279 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 280 |  |  |  |  |  |  |  |
| 281 | dfA.72 | 成龙 | 0 | 1983 | 6 | 28 |  |
| 282 | dfB.72 | 成龍 | 0 | 1983 | 6 | 28 |  |
| 283 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 284 |  |  |  |  |  |  |  |
| 285 | dfA.73 | 江澤民 | 0 | 1958 | 11 | 23 |  |
| 286 | dfB.73 | 江泽民 | 0 | 1958 | 11 | 23 |  |
| 287 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 288 |  |  |  |  |  |  |  |
| 289 | dfA.74 | 港督 | 0 | 1985 | 2 | 27 |  |
| 290 | dfB.74 | 香港總督 | 0 | 1985 | 2 | 27 |  |
| 291 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 292 |  |  |  |  |  |  |  |
| 293 | dfA.75 | 李世民 | 0 | 1989 | 3 | 11 |  |
| 294 | dfB.75 | 唐太宗 | 0 | 1989 | 3 | 11 |  |
| 295 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 296 |  |  |  |  |  |  |  |
| 297 | dfA.76 | 宋慶齡 | 0 | 1986 | 4 | 15 |  |
| 298 | dfB.76 | 宋庆龄 | 0 | 1986 | 4 | 15 |  |
| 299 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 300 |  |  |  |  |  |  |  |
| 301 | dfA.77 | 華佗 | 1 | 1986 | 2 | 4 |  |
| 302 | dfB.77 | 华佗 | 1 | 1986 | 2 | 4 |  |
| 303 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 304 |  |  |  |  |  |  |  |
| 305 | dfA.78 | 董赵洪娉 | 1 | 1978 | 12 | 26 |  |
| 306 | dfB.78 | 董趙洪娉 | 1 | 1978 | 12 | 26 |  |
| 307 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 308 |  |  |  |  |  |  |  |
| 309 | dfA.79 | 拿破侖 | 1 | 1972 | 11 | 17 |  |
| 310 | dfB.79 | 拿破仑一世 | 1 | 1972 | 11 | 17 |  |
| 311 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 312 |  |  |  |  |  |  |  |
| 313 | dfA.80 | 长春真人 | 1 | 1968 | 9 | 10 |  |
| 314 | dfB.80 | 丘处机 | 1 | 1968 | 9 | 10 |  |
| 315 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 316 |  |  |  |  |  |  |  |
| 317 | dfA.81 | 孫武 | 1 | 1985 | 10 | 31 |  |
| 318 | dfB.81 | 孙武 | 1 | 1985 | 10 | 31 |  |
| 319 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 320 |  |  |  |  |  |  |  |
| 321 | dfA.82 | 蘇格拉底 | 0 | 1978 | 5 | 23 |  |
| 322 | dfB.82 | 苏格拉底 | 0 | 1978 | 5 | 23 |  |
| 323 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 324 |  |  |  |  |  |  |  |
| 325 | dfA.83 | 耶蘇 | 0 | 1976 | 5 | 9 |  |
| 326 | dfB.83 | 耶稣 | 0 | 1976 | 5 | 9 |  |
| 327 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 328 |  |  |  |  |  |  |  |
| 329 | dfA.85 | 女人 | 0 | 1979 | 2 | 16 |  |
| 330 | dfB.85 | 女性 | 0 | 1979 | 2 | 16 |  |
| 331 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 332 |  |  |  |  |  |  |  |
| 333 | dfA.86 | 郑成功 | 1 | 1978 | 4 | 14 |  |
| 334 | dfB.86 | 鄭成功 | 1 | 1978 | 4 | 14 |  |
| 335 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 336 |  |  |  |  |  |  |  |
| 337 | dfA.87 | 尉迟敬德 | 1 | 1992 | 2 | 17 |  |
| 338 | dfB.87 | 尉迟恭 | 1 | 1992 | 2 | 17 |  |
| 339 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 340 |  |  |  |  |  |  |  |
| 341 | dfA.88 | 川岛芳子 | 0 | 1976 | 8 | 15 |  |
| 342 | dfB.88 | 川島芳子 | 0 | 1976 | 8 | 15 |  |
| 343 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 344 |  |  |  |  |  |  |  |
| 345 | dfA.89 | 唐僧 | 0 | 1983 | 11 | 9 |  |
| 346 | dfB.89 | 唐三藏 | 0 | 1983 | 11 | 9 |  |
| 347 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 348 |  |  |  |  |  |  |  |
| 349 | dfA.90 | 李柱铭 | 1 | 1966 | 4 | 4 |  |
| 350 | dfB.90 | 李柱銘 | 1 | 1966 | 4 | 4 |  |
| 351 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 352 |  |  |  |  |  |  |  |
| 353 | dfA.91 | 鈴木善幸 | 1 | 1971 | 7 | 1 |  |
| 354 | dfB.91 | 铃木善幸 | 1 | 1971 | 6 | 1 |  |
| 355 | agreement pattern: | 2 | 2 | 2 | 0 | 2 | 0.976 |
| 356 |  |  |  |  |  |  |  |
| 357 | dfA.92 | 小泉純一郎 | 1 | 1976 | 7 | 16 |  |
| 358 | dfB.92 | 小泉纯一郎 | 1 | 1976 | 7 | 16 |  |
| 359 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 360 |  |  |  |  |  |  |  |
| 361 | dfA.93 | 宋美龄 | 0 | 1990 | 12 | 19 |  |
| 362 | dfB.93 | 宋美齡 | 0 | 1990 | 12 | 19 |  |
| 363 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 364 |  |  |  |  |  |  |  |
| 365 | dfA.94 | 胡錦濤 | 0 | 1999 | 1 | 13 |  |
| 366 | dfB.94 | 胡锦涛 | 0 | 1999 | 1 | 13 |  |
| 367 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 368 |  |  |  |  |  |  |  |
| 369 | dfA.95 | 帝尧 | 1 | 1979 | 2 | 27 |  |
| 370 | dfB.95 | 尧 | 1 | 1979 | 2 | 27 |  |
| 371 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 372 |  |  |  |  |  |  |  |
| 373 | dfA.96 | 介之推 | 0 | 1981 | 8 | 7 |  |
| 374 | dfB.96 | 介子推 | 0 | 1981 | 8 | 7 |  |
| 375 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |
| 376 |  |  |  |  |  |  |  |
| 377 | dfA.98 | 李鸿章 | 1 | 1970 | 12 | 20 |  |
| 378 | dfB.98 | 李鴻章 | 1 | 1970 | 12 | 20 |  |
| 379 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 380 |  |  |  |  |  |  |  |
| 381 | dfA.99 | 梁啟超 | 1 | 1966 | 2 | 17 |  |
| 382 | dfB.99 | 梁启超 | 1 | 1966 | 2 | 17 |  |
| 383 | agreement pattern: | 2 | 2 | 2 | 2 | 2 | 1 |
| 384 |  |  |  |  |  |  |  |
| 385 | dfA.100 | 清世祖 | 1 | 1968 | 7 | 4 |  |
| 386 | dfB.100 | 顺治帝 | 1 | 1968 | 7 | 4 |  |
| 387 | agreement pattern: | 0 | 2 | 2 | 2 | 2 | 0.9994 |

1. Relative performance and expected computation time of available methods

During our methods development, we explored a range of approaches to Chinese name matching, ranging from single pairs of transformations and similarity metrics, to a simple linear combination of multiple transformation-similarity pairs, to large ensembles of boosted decision trees. In general, computation time and name-matching performance increased with model complexity, and so the user may wish to consider more complex models for smaller linkage applications (or for larger ones if they are willing to wait a while for results), and perhaps accept slightly diminished performance for the sake of computation efficiency in other scenarios.

Here we plot the relative performance on non-exact matches for each method



Below we plot the expected runtime for the best example of each method on a computer with 24 cores working in parallel as record sizes increase:

