Aiding Address-Based Matching Through Building Name Standardization

Census and Statistics: Innovations in U.S. Census Bureau Geographic Systems
ESRI User Conference
Wednesday, July 12, 2017

Kevin Holmes
Geographer
Address Standards, Criteria, and Quality Branch (ASCQB): Geography Division
U.S. Census Bureau

United States Census Bureau
U.S. Department of Commerce
Economics and Statistics Administration
U.S. CENSUS BUREAU
census.gov

This presentation does not contain Census Title 13 information

Census Addressing

Master Address File (MAF)

- Address repository for Censuses and Surveys
- MAF/TIGER database (relational)
- Developed for 2000 Census, maintained throughout decades

Group Quarters (GQ) Addresses

- Specific living/service arrangements
  - Prisons, nursing homes, college dormitories, etc.
- Different enumeration methodology
- Additional attribution: Facility or Building Name
  - **999 University Blvd., 12345 -> 'Redlands Residence Hall'**

This presentation does not contain Census Title 13 information

Matching Abstract

Administrative Datasets

- Authoritative resources for continual MAF improvement (e.g. tribal, state, local governments, USPS Delivery Sequence File, etc.)
- MAF record matching primarily address-driven
- Can additional information be used to aid matching?

GQ Name Standardization and Matching (SAM)

- Aiding address matching for datasets with:
  - Missing/Incomplete address elements
  - Non-city style addresses (P.O. or R.R. boxes)
  - Data variations not conducive to typical address-based matching
  - Additional attribution available

This presentation does not contain Census Title 13 information

Among Other Challenges…

MAF and administrative GQ names may also differ

- SPRINGFIELD CO CORR CTR
- SHELBYVILLE UNIV. VILLAGE RES HALL
- GENERAL HOSP. EXT. CARE
- SMITHERS GRP HME

Name Standardization to Facilitate Matching

SPRINGFIELD CO CORR CTR	->	SPRINGFIELD COUNTY CORRECTIONAL CENTER
SHELBYVILLE UNIV. VILLAGE RES HALL	->	SHELBYVILLE UNIVERSITY VILLAGE RESIDENCE HALL
GENERAL HOSP. EXT. CARE	->	GENERAL HOSPITAL EXTENDED CARE
SMITHERS GRP HME, INC.	->	SMITHERS GROUP HOME, INCORPORATED

This presentation does not contain Census Title 13 information

Standardization

Standardization affects match score:

RAW NAMES		
MAF GQ NAME	**Local GQ Name**	**Score**
HOLMES CO COMM COLLEGE	*HOLMES COUNTY COMM. COL.*	0.85
STANDARDIZED NAMES		
MAF GQ NAME	**Local GQ Name**	**Score**
HOLMES COUNTY COMMUNITY COLLEGE	*HOLMES COUNTY COMMUNITY COLLEGE*	1

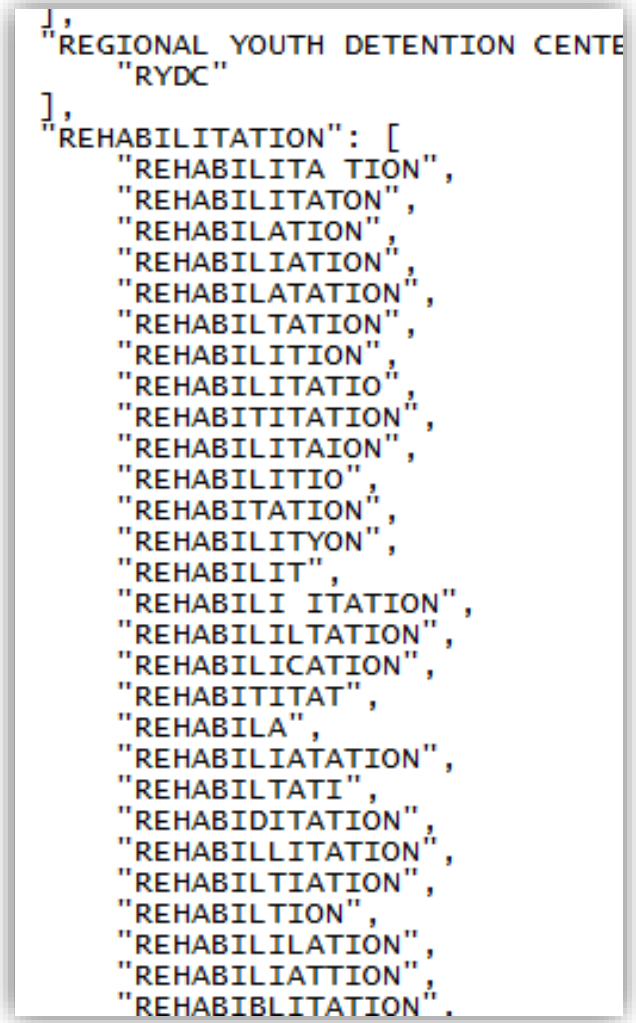
Standardization

Word replacement dictionaries

- Multiple JSON file(s)
- Replace ‘bad’ version of word with ‘good’ version

‘BAD’	‘GOOD’
APTS.	APARTMENTS
HOSP.	HOSPITAL
UNIV	UNIVERSITY

Current standardization replacement dictionary includes ~1,000 substitutions



This presentation does not contain Census Title 13 information

Substitution: Not So Fast

'CO': COUNTY? COLORADO?

'ST': STATE? STREET? SAINT?

During standardization:

- Iterate through each replacement word 'option' within the GQ name
  - If both MAF & local GQ name cite same 'multi-replacement' string, do not iterate/replace
- Compare source & MAF GQ name strings
- String with best match score *may* be best match

RAW NAMES		
HOLMES CO COMM COLLEGE	*HOLMES COUNTY COMM. COL.*	
STANDARDIZED NAMES		
MAF GQ NAME	**Local GQ Name**	**Score**
HOLMES COUNTY COMMUNITY COLLEGE	*HOLMES COUNTY COMMUNITY COLLEGE*	1
HOLMES COLORADO COMMUNITY COLLEGE	*HOLMES COUNTY COMMUNITY COLLEGE*	0.895

Multi-replacement substitution

- Easier said than done...

'JEFFERSON CTY CO EXT. CARE FAC ':

'JEFFERSON COUNTY COUNTY EXTENSION CARE FACILITY',
'JEFFERSON CITY COUNTY EXTENSION CARE FACILITY',
'JEFFERSON COUNTY COLORADO EXTENSION CARE FACILITY',
'JEFFERSON CITY COLORADO EXTENSION CARE FACILITY',
'JEFFERSON COUNTY COUNTY EXTENDED CARE FACILITY',
'JEFFERSON CITY COUNTY EXTENDED CARE FACILITY',
'JEFFERSON COUNTY COLORADO EXTENDED CARE FACILITY',
'JEFFERSON CITY COLORADO EXTENDED CARE FACILITY'

```
#Function to take single standardized name output and report multi-rep combinations
def multiRep(stanNme):
    #load in multireps
    with open(r'\\directory.json') as mJsonIn:
        subDictMJson = json.load(mJsonIn)
    origg = stanNme.keys()[0]
    standd = stanNme.values()[0]
    words = standd.split(' ')
    numWords = len(words)
    if numWords == 1:                                                    #if only one word in input string
        mrs = {}                                                  #CHANGE TO REPORT OUT AS SINGLE ITEM LIST???
        if standd in subDictMJson.keys():
            reps = []
            for gd in subDictMJson[standd]:
                reps.append(str(gd))
            mrs[origg] = reps
        else:
            sper = []
            sper.append(standd)
            mrs[origg] = sper
        return mrs
    if numWords > 1:                                                     #if >1 word in input string
        mwMrs = {}
        strbld = []
        for word in words:
            if word not in subDictMJson.keys() and len(strbld) == 0:     #if not a multirep word and it's the first word
                bld1 = []
                bld1.append(word)
                strbld.append(bld1)
                continue
            if word not in subDictMJson.keys() and len(strbld) > 0:      #if not a multirep word and other words before
                for s in strbld:
                    s.append(word)
            if word in subDictMJson.keys() and len(strbld) == 0:         #if a multirep word and it's the first word
                for wd in subDictMJson[word]:
                    bld2 = []
                    bld2.append(str(wd))
                    strbld.append(bld2)
                continue
            if word in subDictMJson.keys() and len(strbld) > 0:          #if a multirep word and other words before it
                gds = subDictMJson[word]
                gdsCnt = len(gds)
                stbldOrig = []
                for sit in strbld:
                    stbldOrig.append(sit)
                for st in stbldOrig:
                    origSt = []
                    for i in st:
                        origSt.append(i)
                    st.append(str(gds[0]))
                    for gd in gds[1:]:
                        newbld = []
                        for o in origSt:
                            newbld.append(o)
                        newbld.append(str(gd))
                        strbld.append(newbld)
                        continue
        #return strbld                      #reformat, returning as dictionary items
        strbldLst = []
        for entr in strbld:
            strbldSt = ''
            strbldSt = ' '.join(entr)
            strbldLst.append(strbldSt)
        mwMrs[origg] = strbldLst
    return mwMrs
```


This presentation does not contain Census Title 13 information

Matching – Exact & Equivocated

After standardization, GQ Names may still not match exactly

- Equivocated matching to 'close enough' strings
- Jaro-Winkler (JW) string similarity scoring
  - Used in street name equivocation

0.0 >>>>>>>>>>>>>>>>>>>>>>>> **1.0**

No Match — Equivocated Match — Exact Match

This presentation does not contain Census Title 13 information

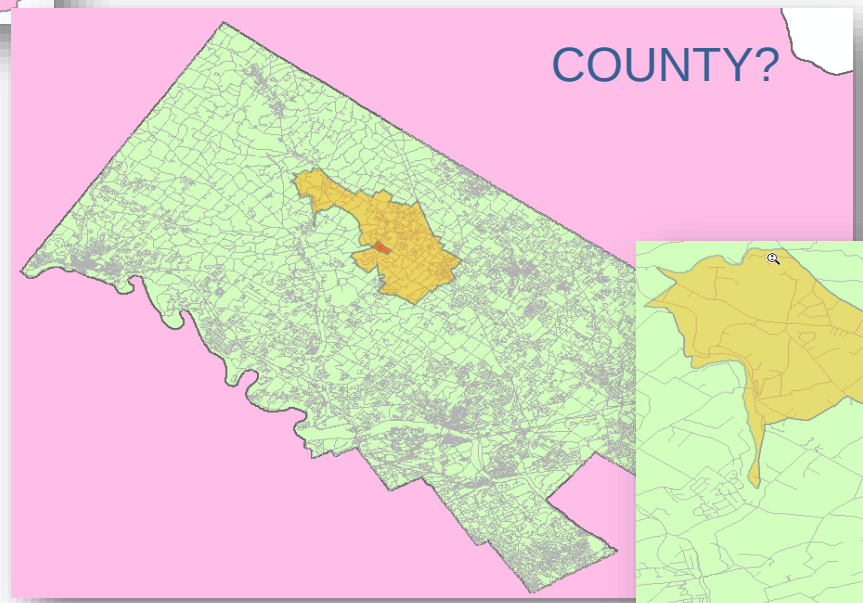
Matching - Examples

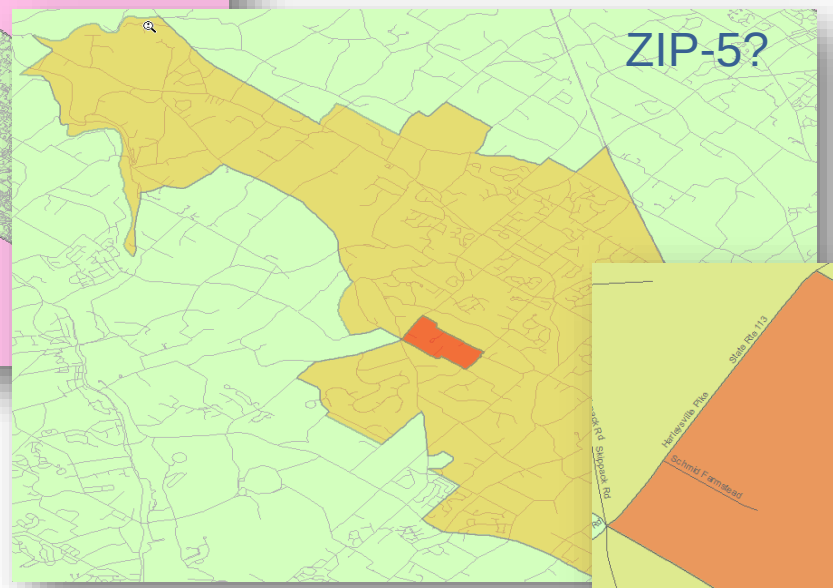
MAF GQ NAME	SOURCE GQ NAME	JW SCORE
CENSUS NURSING CENTER	CENSUS NURSING CENTER	1
MAF HEALTH CARE CENTER	MAF HEALTHCARE CENTER	0.973
TIGER REHABILITATION CENTER WEST GERIATRIC UNIT	TIGER REHABILITATION NURSING CENTER	0.902

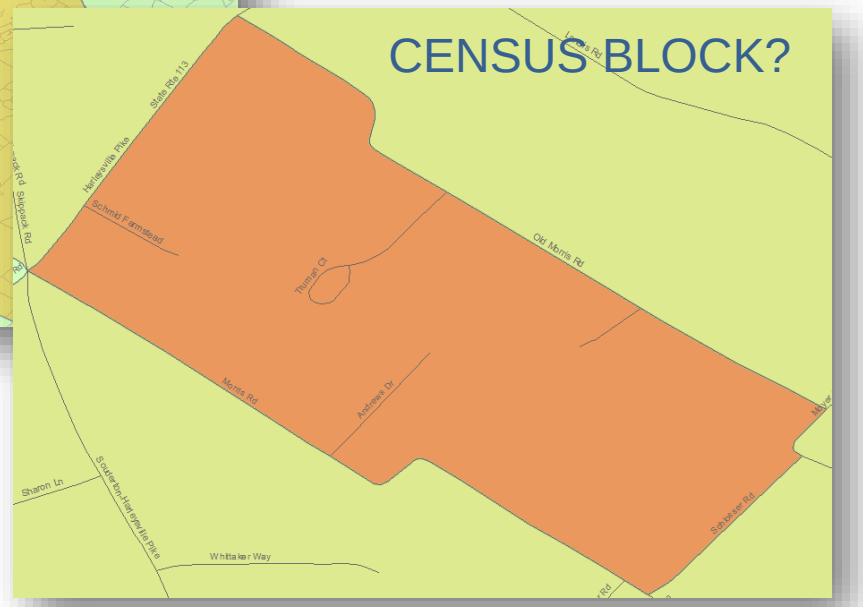
Blocking

- Source datasets may span the nation
  - 1 vs. national record matching not efficient
  - Multiple matches may exist across nation
- Compartmentalizing source and MAF datasets increase efficiency/accuracy
  - Too large, processing run time becomes a factor
  - Too small, ‘match candidate’ universes too small

Blocking







- Direct attribution
  - File fields/values
- Spatial interpolation
  - Derived from source X/Y (if available)

This presentation does not contain Census Title 13 information

```
mBlock = []   #keeping generic...could be zips or STCOU codes
mBlockStr = ''


if blockING == 'ZIP':
    #Create list of Source zips
    print 'Creating list of unique source zips...'
    for i in mtch[1:]:
        zipp = i[2]
        if zipp not in mBlock and zipp != None:
            mBlock.append(zipp)
            mBlockStr+=zipp+','
```

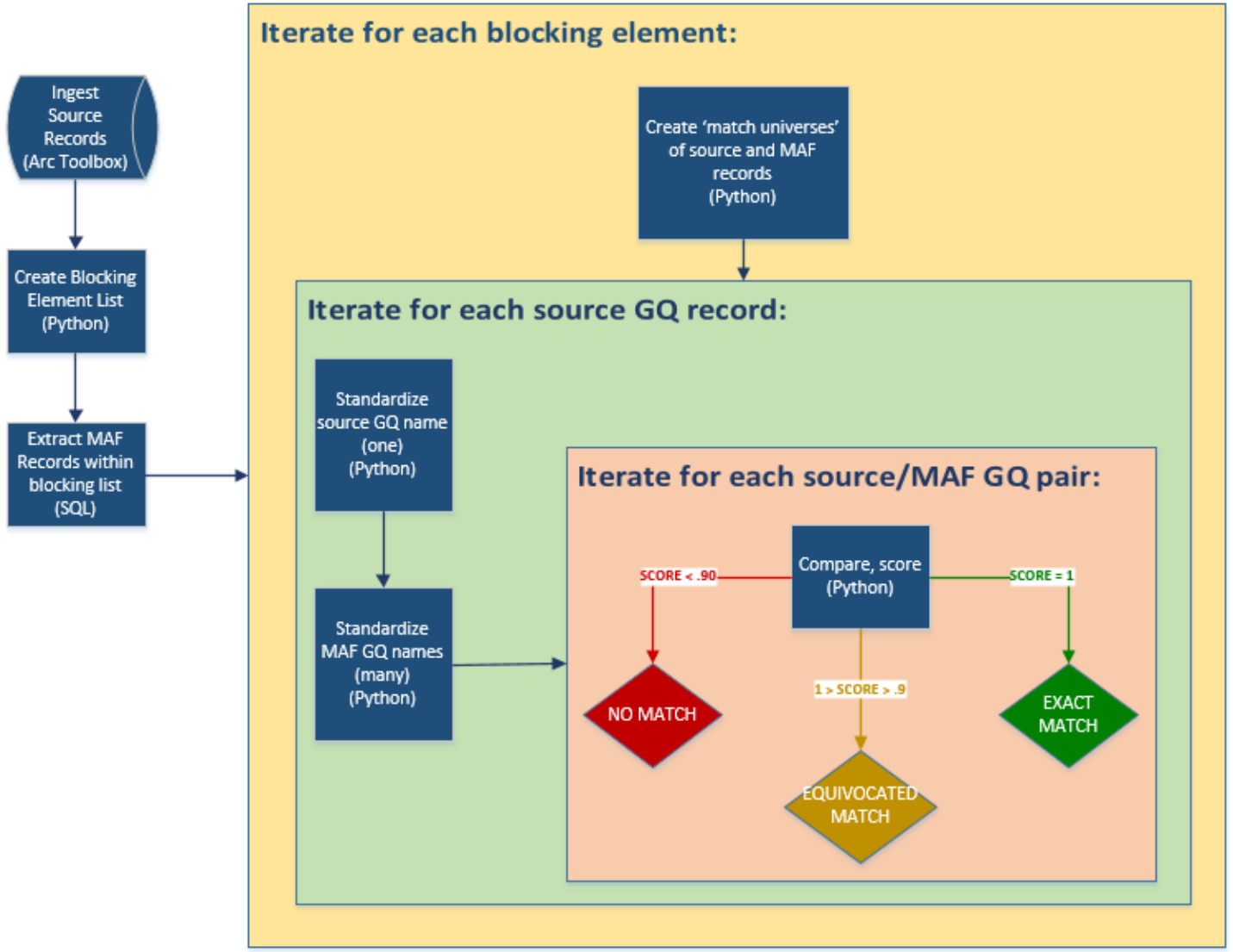
Creates blocking list of values…

…passes blocking values as SQL selection parameter

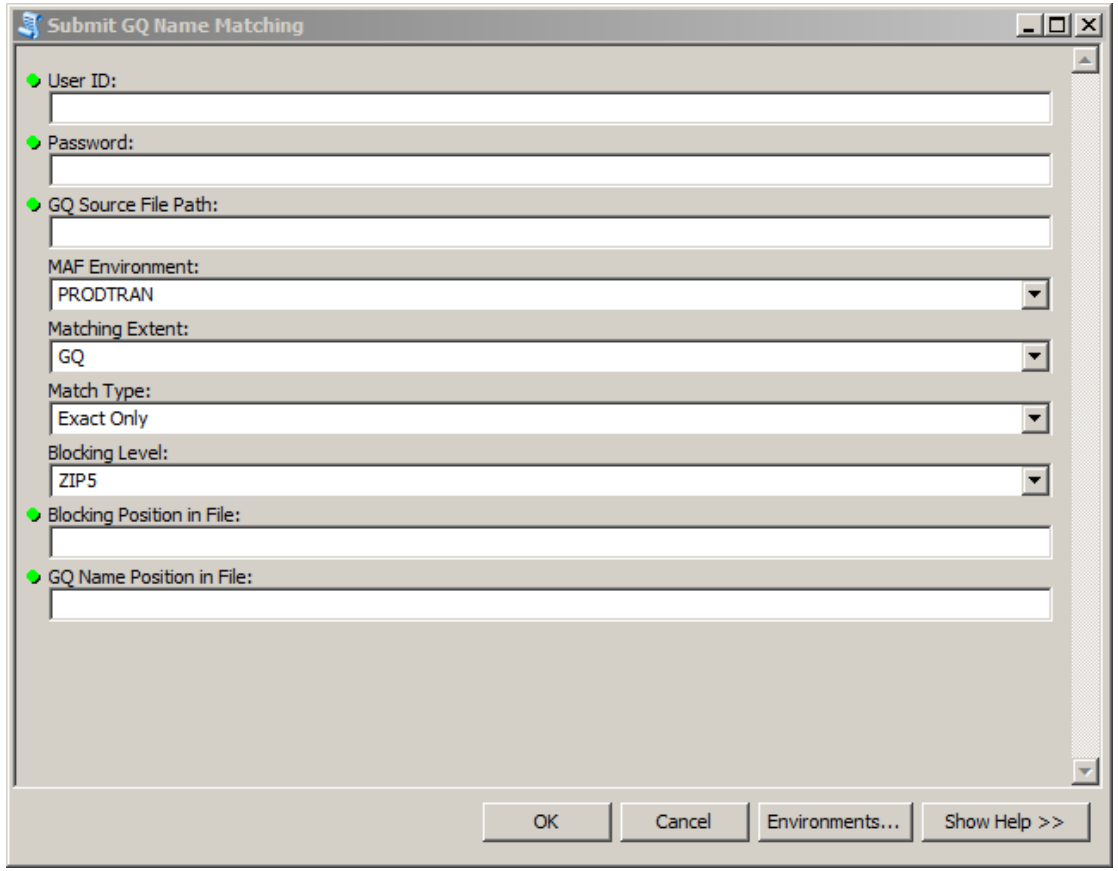
```
#ORACLE PULLS
#connect to environment and pull distinct GQ & SP names
db = cx_Oracle.connect(userName, passWord, env)
query = db.cursor()


#create custom sql 'where' clause based on blocking type
if blockING == 'ZIP':
    sqlWhere = 'and c.zip in ('+mBlockStr[:-1]+')'

if blockING == 'STCOU':
    sqlWhere = 'and b.statefp||b.countyfp in ('+mBlockStr[:-1]+')'



#PULLING MAF GQNAMES
print '\nRunning GQ Name sql on '+env+' where '+blockING+' matches...'
query.execute('''
select 'things'
from table a
join table b on table a.value = table b.value
join table c on table b.oid = table c.value
join maftiger.table d on table c.value = table d.value
where a.value is not null
'''+sqlWhere+'''
and (table c.value = 'Y' or table c.value = 'Y')
order by table a.value
''')
```

Implementation

GQ SAM = Blocking + Standardization + Matching



Initialization – ArcMap Toolbox



RESULTS REPORTED TO ORACLE TABLES FOR POST-MATCH PRIORITIZATION

Matching Examples: Good

- Exact match, no standardization
  - SOURCE: BREMEN NURSING HOME
  - MAF: BREMEN NURSING HOME

  Match score: 1.0

- Exact match, with Standardization
  - SOURCE: WEST PARK REHAB AND NURSING CENTER
  - MAF: WEST PARK REHABILITATION AND NURSING CENTER

  WEST PARK REHABILITATION AND NURSING CENTER

  Match score: 1.0

Matching Examples: Probably?

Equivocated (.9 < score > 1.0)

- SOURCE: LIFE CARE CENTER
- MAF: LIFE CARE CENTER OF ATLANTIS

Match score: 0.914

- SOURCE: KEVIN HOLMES RESIDENCE HALL
- MAF: K. HOMES RESIDENCE HALL

Match score: 0.906

Matching Examples: Probably Not

Equivocated (.9 < score > 1.0)

- SOURCE: MAJESTIC OAKS – WEST WING
- MAF: MAJESTIC OAKS NORTH

Match score: 0.901

- SOURCE: GOLDEN GIRLS CENTER SPRINGFIELD
- MAF: GOLDEN GIRLS CENTER - SHELBYVILLE

Match score: 0.908

Future iterations may restrict matching of names containing directionals etc.

Matching Examples: No Matches

- Unqualified match score: less than .9
- Non-match does not = missing!
  - Missing/incomplete/incorrect blocking information
  - Incomplete word substitution directory
  - Names simply too different
    - SOURCE: ROLLING HILLS NURSING CENTER AT WAYFIELD
    - MAF: ROLLING HILL REHABILITATION AND NURSING

    Match score: 0.874
- If scoring break is reduced, greater chance of 'bad' matches

Challenges/Issues

- Name standardization replacement dictionary always improving
  - Existing MAF GQ name elements can be evaluated (spell checker)
    - New GQ names could be standardized before MAF ingestion
  - External source datasets a constant unknown

- No ‘best’ matching score break
  - .90 used for our equivocation cut-off
  - Some ‘good’ matches found <.90, but lends to more questionable/bad matches

- IF THE DATASETS ARE TOO DIFFERENT, THE PROCESS DOES NOT WORK ☹

Multiple MAF Matches

- Multiple MAF record matches are common
  - Duplication (new/retired records)
  - GQ vs. Facility name
  - One-to-Many: ‘HOLMES HALL A’ matches to ‘HOLMES HALL 1’, ‘HOLMES HALL 2’, ‘HOLMES HALL B’, etc.

- A ‘match’ does not necessarily mean it’s the **best** match
  - Additional MAF attribution filtering may be used to prioritize better/best matches

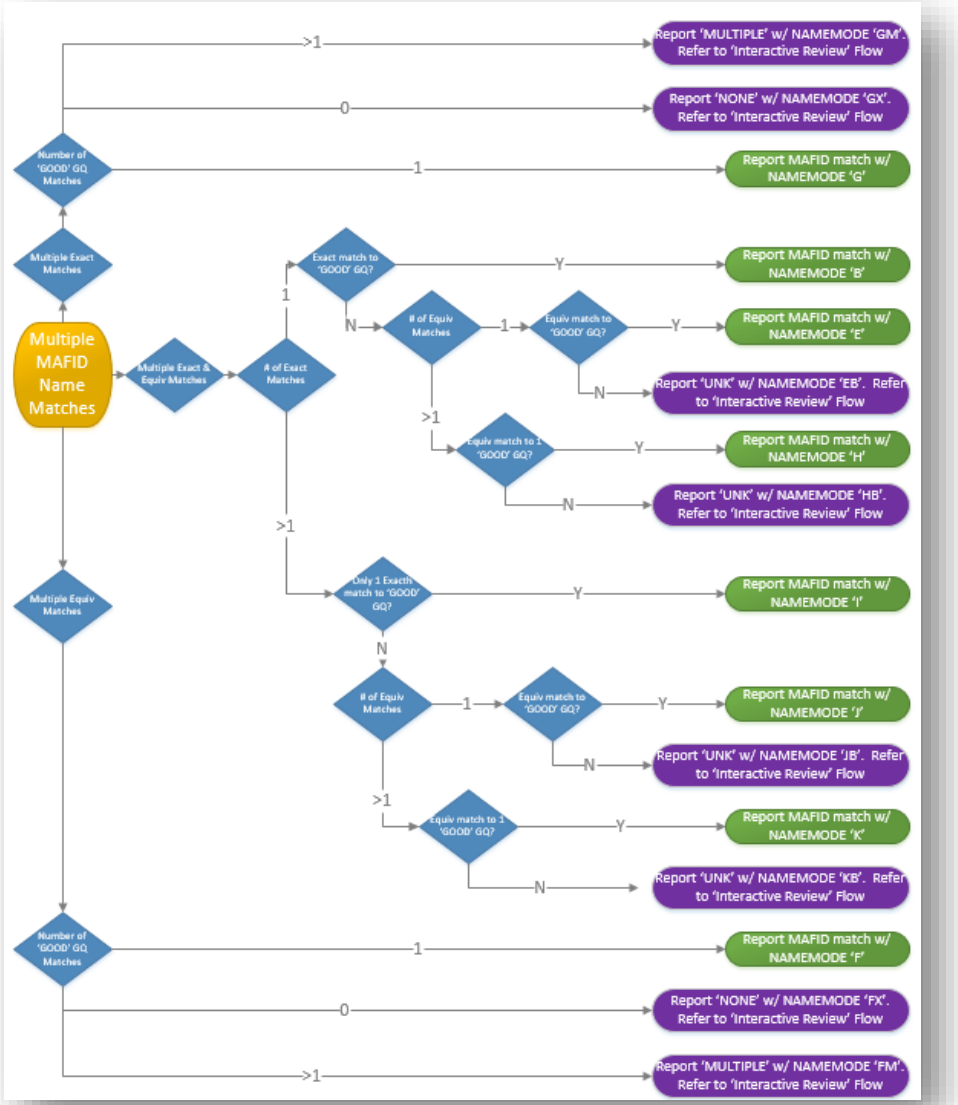
Evaluation Strategies

- For Matches (single or multiple):
  - How many matches?
  - Exact or equivocated?
  - Are matches to 'good' MAF records (recently enumerated, other source verified, etc.)
    - How many? If multiple, which one is best?
    - If 0, manual MAF searching may be required
- For Non-Matches:
  - Manual MAF searching may be required
  - Why couldn't a match be made?
    - Blocking elements different
    - Replacement dictionary lacking?
    - Name too different?
  - Potential MAF adds?

Post-Match Filtering

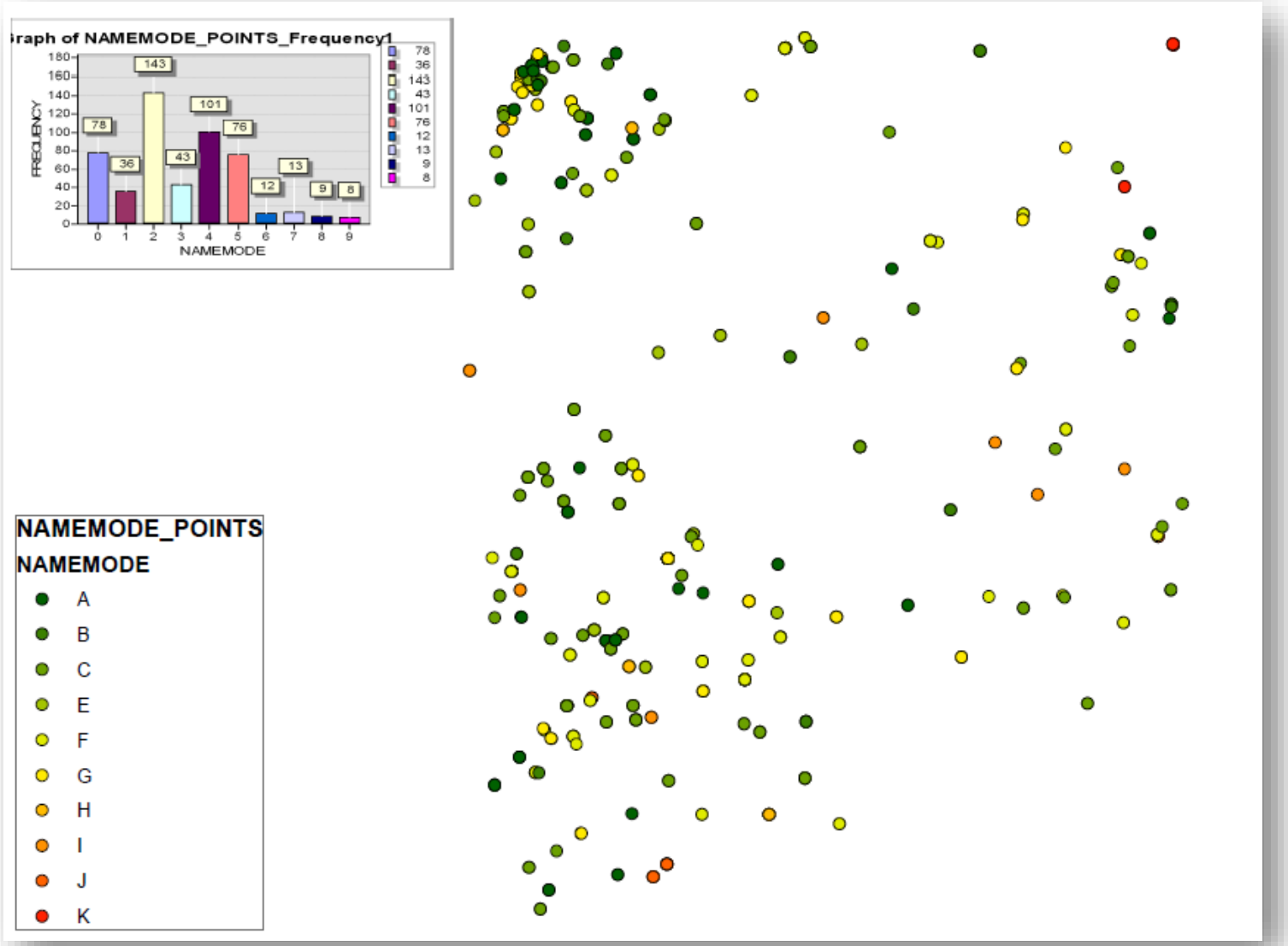
Post-match filtering & selection to choose ‘best’ match, factoring in:

- Number of matches
- Exact or equivocated
- Nature of matched MAF record



This presentation does not contain Census Title 13 information

Interactive Capabilities: ArcMap



Questions?

Kevin Holmes

kevin.j.holmes@census.gov

215.356.6275