Record Linkage Software, Utilities, and Other Fun Stuff

Rich Pinder
Los Angeles Cancer Surveillance Program
rpinder@usc.edu
Files at: www-hsc.usc.edu/~rpinder

Presented at the NAACCR Toolkit Workshop
October 8, 2002

Introduction

- **Wide range of linkage experience here today**
- **Fairly long session**
- **I know what _I_ think is interesting... But...**
- **Don’t leave session with questions we havent discussed!**
- **Break – good time to talk offline – it’ll benefit the group**

Introduction (cont)

- **How many heard of ‘Deterministic’ vs ‘Probabilistic record linkage ?**
- **How many can define them ??**
- **Linkage is broad term**
- **No right/wrong way**
- **80/20 rule –**

Questions submitted:

- Q. Are there guidelines regarding the appropriate weights and/or appropriate variables to be used when linking?
- Q. Is the clerical reviewer back in automatch?
- Q. Would be interesting to hear any experiences that other registries have to share with innovative ways they have performed linkages.
- Q. Our registry is bringing on MCOs (??) on as a reporting source, and it is a whole different process to utilize claims data than registry data, and to match in the claims data to the registry data. Just wondering if any other registries are processing claims data, and what their experiences have been.

Objectives

- **First, some TOOLS**
- **Record Linkage theory**
- **Ascential's Integrity**
- **Open forum – 'linkage', nos**

Programming Editors

- **Ever need to ‘see’ your data ?**
- **Can be used for Data Transformation (ie search/replace & columnar formatting**
- **Current products sophisticasted**
- **Emacs** www.gnu.org/software/emacs/emacs.html
- **Epsilon** www.lugaru.com
- **CRiSP** www.vital.com/
- **BBEdit (Mac OS X)** www.barebones.com

Scripting Languages

- **Great for data manipulation jobs**
- **Open source – wide user base**
- **Code samples galore**
- **News groups super**
- **Good way to expand knowledge**
- **Perl & Python**

Scripting Languages - Python

- **Python 101 – take home utilities**
- **3 scripts:**
  1. **Check an ascii file to be sure its fixed length**
  2. **Append nickname information to a datafile, using a defined list of accepted nicknames (ie Rich = Richard)**
  3. **Append soundex to a datafile**

Scripting Languages - Python

- **Python is a freely available, interpreted, scripting language.**
- **Modern environment – object oriented**
- **Makes a fine ‘first’ language for beginning programmers**
- **Text scripts (default .py) extension are byte optimized (converted to binary – not compiled) into a .pyc file first time run**
- **Examples today character based – but Python has a multi platform gui toolkit based on Tcl**
- **Great resources (including introductions)**

  www.python.org
- **Great book: Python, by Chris Fehily**

Scripting Languages - Python

```
import sys, os

#.........................
# Define Functions first

def TestLength():

    sml = 0
    big = 0
    counter = 0
    print "\n\nFixed Length Line Tester:\n"
    for line in f1.readlines() :
        counter = counter + 1
        lil = len(line) - 1
        if (sml == 0) :
            sml = lil.................
```

Works with modules

Comments start with #

Defining functions

No line termination characters

Assignment (no declaration)

Control Structures use indentation

Scripting Languages - Python

```
# Body of Program begins here

if len(sys.argv) < 2 :
    print "\n\nUsage: python linetest.py <filename>"
    sys.exit()

thefile = sys.argv[1]

f1  =  open(thefile)

TestLength()

f1.close
```

Command Line Arguments

f1 – new object

Calling function

Scripting Languages - Python

nicki.py features:

- **Checks command line arguments**
- **Reads nicknames and standardized names into a 'dictionary' (a python structure)**
- **Searches on key – appends either standardized name (if key found), or original name (if not found)**
- **Redirects STDOUT (standard out) to file object**

Scripting Languages - Python

addsndx.py features:

- **Calls the soundex algorithm from the an external Module (soundex.py)**

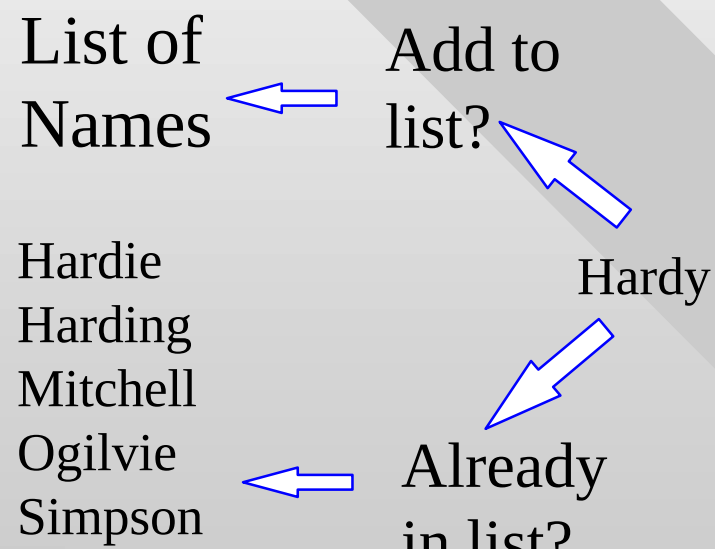
Deterministic and Probabilistic Record Linkage Methods

Denver, October 2002

Why Link Records?

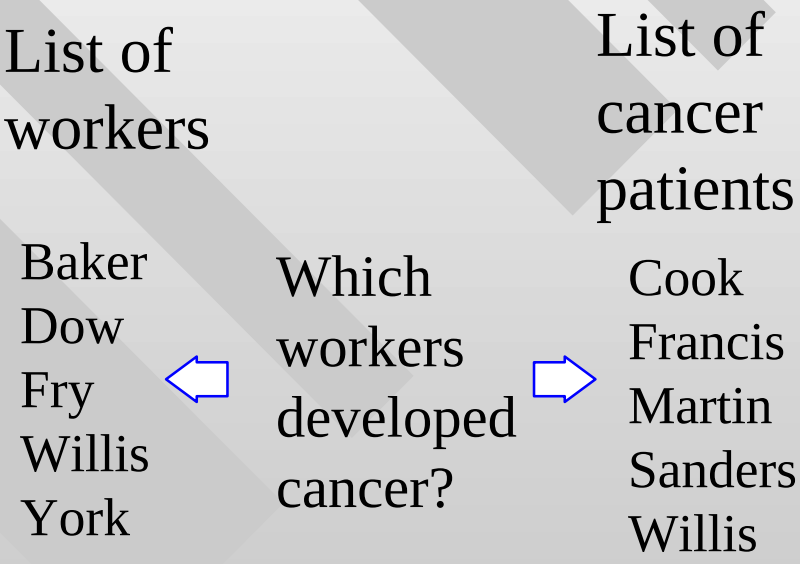
1) Registry Operations

Because you have a Master List and wish to add new names to it.



2) Research Linkages

Because you have two lists and wish to compare them.



- **Real Time environment – desirable**
  - **Mimics work flow**
  - **Time/sequence advantage over Batch**

- **Integrated vs symbiotic**
  - **Sophistication vs ease of implementation**
  - **Can your Database environment sustain?**

- **‘Home Grown’ Fine for production ?**
  - **Simplified algorithm (Deterministic ok?)**
  - **Requires increased Database index/keys resources?**
  - **80/20 rule – will it suffice ?**
  - **Black Box story**

- **Third Party products advantages**
  - **Better algorithms ? (Probabilistic, Complex comparators)**
  - **Easier to document and defend?**
  - **No maintenance**
  - **Concurrency issues**

- **Production**
  - **Follow up:**
    - Mortality: State vital stats; SSA DMF;
    - Voter Registration;
  - **Work Process Flow:**
    - Pathology review
    - New case additions
    - Unduplication
- **Research**
  - **Incidence:**
    - Cohort studies
    - Aids linkages
    - Worker effects – Aircraft workers

Topics to consider

- **Code consistency in your data**
- **File Standardization & File review**
  - **Look for problems/undocumented issues in data**
  - **Is coding consistent**
  - **Review data manually – beware of formatting errors**
  - **How much missing data ?**
  - **Know accuracy of elements**

"Exact Match" / *Deterministic* Linkage

- Simpler method of matching.
- Records agreeing "exactly" within an individual data field or a group of common fields between records.
- Approach relies on files having unique identifying information
  - health insurance number, social security number,surnames, given names
    - » minimal amount of missing or erroneous information

“Exact Match” / *Deterministic* **Linkage**

- Primary advantages**:**
  - **technique brings together record pairs very efficiently, simply by sorting both files using a common unique identifier as the key field.**
  - **can be successfully applied when accurately recorded unique personal identifying information is available**

"Exact Match" / *Deterministic* Linkage

- Primary disadvantages:
  - **absence / incompleteness / inaccuracy of key identifying variables**
    - » **e.g., inconsistencies from record to record in the accuracy of surnames, given names and other identifiers, such as birth date.**
  - **spelling and transcription errors at time of data collection**
  - **use of nicknames and proper names used interchangeably; name changes over time (marriage/adoption)**

"Exact Match" / *Deterministic* Linkage

- **Develop rules based on variables present on both files e.g.,** matches **if any of these conditions are met:**
  1. **same surname, 1st name, ID#, date of birth or**
  2. **same surname, 1st name, date of birth or**
  3. **same surname, 1st name initial, ID#, age, etc.**
  - **Note: there are 2^n possible patterns of agreement and disagreement on *n* fields:**
    - **e.g., 10 fields = 2^{10} = 1,024 possible combinations of fields agreeing and disagreeing!**

"Exact Match" / *Deterministic* Linkage

- **This doesn't account for missing values and partial agreements.**

- **Specialized code for deterministic combinations often takes years to develop and never quite fulfills its goals. In addition, flexibility is lost.**

Probabilistic Record Linkage

- **Recommended over traditional deterministic methods (i.e. exact matching) methods when:**
  - *coding errors*, *reporting variations*, *missing data* **or** *duplicate records* **encountered by registry**
- **Estimate probability / likelihood that two records are from the same person versus not**
- **Frequency Analysis of data values involved (and IMPORTANT)**

*Probabilistic Linkage **(cont'd)***

- **Landmark papers in computerized probabilistic record linkage by several Canadians in 1960s and 1970s (Fellegi & Sunter, Newcombe, Howe)**
- **Statistics Canada (in collaboration with NCIC) - developed the Generalized Iterative Record Linkage System - GIRLS (based on Fellegi-Sunter model)**
  - **Details in: Newcombe HB. Handbook of Record Linkage. Oxford University Press, 1988**

*Probabilistic Linkage **(cont'd)***

- **Frequency Analysis – examples:**
  - **How common is the surname 'Takaharu' in the Northern Texas Regional Cancer Registry?**
  - **How common is the surname 'Takaharu' in the Tokyo Cancer Registry ?**
  - **If you've got an 'iffy' match – and the Surname is 'Rumplepinder' – you likely to take it ?? (say ssn is missing, and mo/day of birth is wrong)**
  - **If you've got the same 'iffy' match – and the Surname is 'Jones' ???**

*Probabilistic Linkage **(cont'd)***

- **Frequency Analysis – examples:**
  - **You're matching your Cancer file with the Mortality file. What are the impacts of a pair of 'John M Smith' matching with month/yr agreement on birth of 10/23….. Vs the same scenario but an agreement of birth of 10/79**

- **This is a HUGE component of probability**

Probabilistic ***Linkage (cont'd)***

- **Formalization of intuitive concepts regarding outcomes of comparison of personal identifiers**
  - agreement **argues** for linkage **and *disagreement against* linkage**
  - **partial agreement is less strong than full agreement in supporting linkage**
    - **some types of partial agreements are stronger than others (e.g., truncated rare surname vs residence county code)**

Probabilistic ***Record Linkage (cont'd)***

- **Agreement on an uncommon value argues more *strongly* for linkage than a common value (e.g., surname Drazinsky vs Smith)**
- **Agreement on a more specific attribute argues more strongly for linkage than agreement on a less specific one (e.g, SSN # vs sex variable)**
- **Agreement on more attributes, disagreement on few, supports linkage**

Probabilistic ***Record Linkage (cont'd)***

- Blocking**:**
  - **probabilistic linkage step that reduces the number of record comparisons between files**
  - **records for the two files / single file to be linked partitioned into mutually exclusive and exhaustive blocks**
  - **comparisons subsequently made *within* blocks**
  - **implemented by "sorting" the two files by one or more identifying variables**
  - **GREAT analogy: Blocking is like separating your socks into piles based on Color, BEFORE you sort your socks!**

*Probabilistic Linkage **(cont'd)***

The Matching Algorithm can be summarized as follows **(from p 37 of SuperMatchConcepts) 2-14** *INTEGRITY Data Reengineering Environment SuperMATCH Concepts and Reference Guide*

- A block of records is read on both files.
- For each possible record pair in the block, all fields are compared and a composite weight is computed. A matrix of composite weights results. The matrix size is *n*X*m*, where *n* is the number of A records in the block and *m* is the number of B records in the block. The elements of the matrix are the composite weights.
- A Linear Sum Assignment Program is used to optimally assign the best matches.

*Probabilistic Linkage **(cont'd)***

The Matching Algorithm (cont):

- The assigned elements are examined. If they have a weight greater than the cutoff values, they are matched or considered clerical review pairs.
- Duplicates are detected on both files by examining the row and column of an assigned pair. If there is more than one element whose weight is greater than the cutoff weight, it is a potential duplicate.
- The assignments are written out to special pointer files.
- The residual pointers are updated to indicate which records did not match.

Probabilistic Record Linkage ***(cont'd)***

- **Once comparisons within blocks are made:**
  - *weight* **calculated for each field comparison, and total weight derived by summing these separate field comparisons across all fields that have identifying value**
    - **e.g., surname, given names, birth date**
- **Define thresholds for automatically accepting and rejecting a link**
  - **gray area / marginal links reviewed manually**

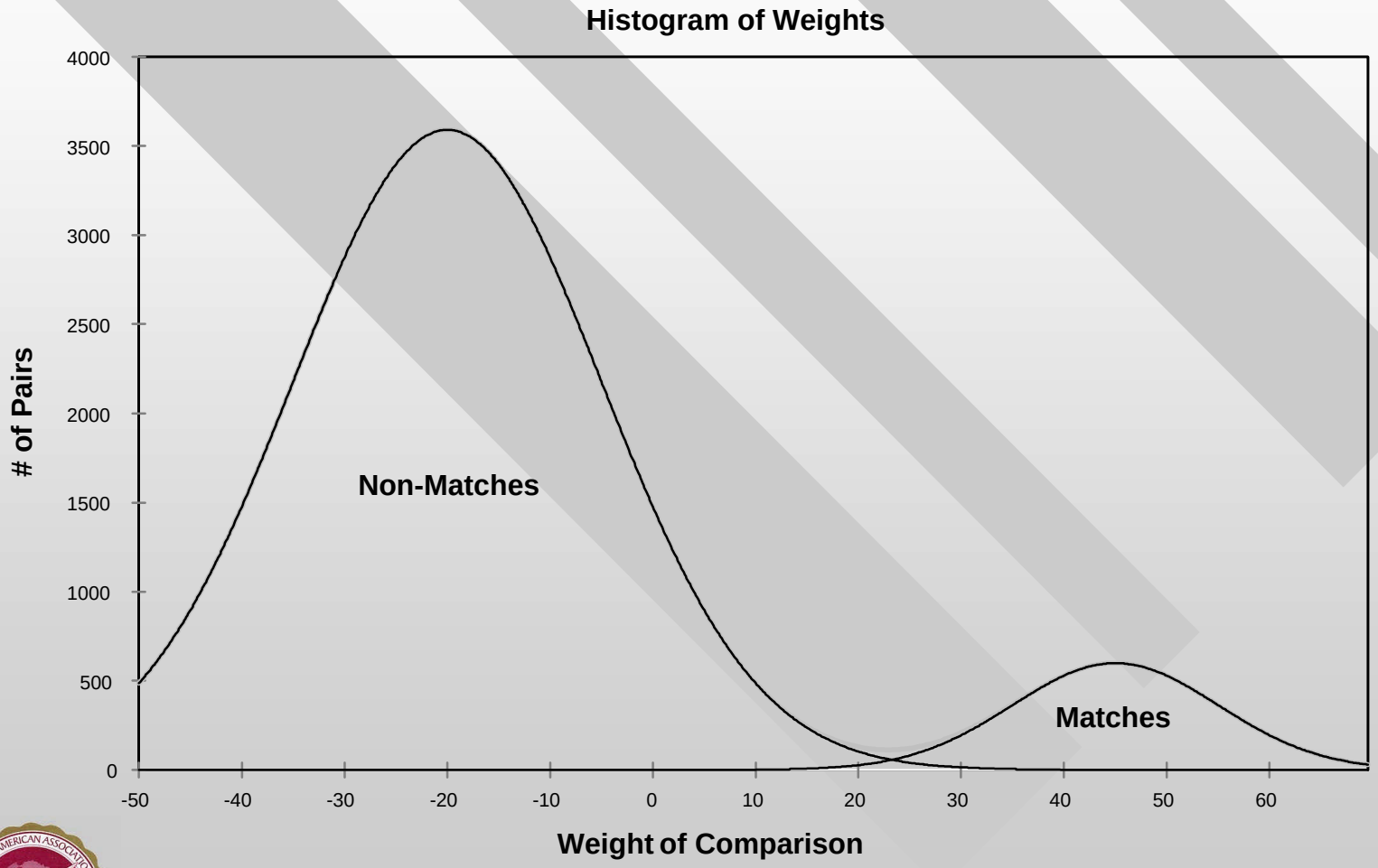
Definition of Weight (Fellegi-Sunter model)

- **Each variable / field has an agreement and a disagreement weight associated with it.**
- **The agreement weight is log (m/u).**
- **The disagreement weight is log ((1-m)/(1-u))**
- **m is the probability that a field agrees given a correctly matched pair (measures the reliability of a field).**
- **u is the probability that a field agrees given a non-matched pair (ie, chance of accidental agreement)**
- **Logarithms are to the base two.**
- **The agreement weight is applied to the field if it matches in the record pair being examined, else the disagreement weight is applied.**

Discrimination

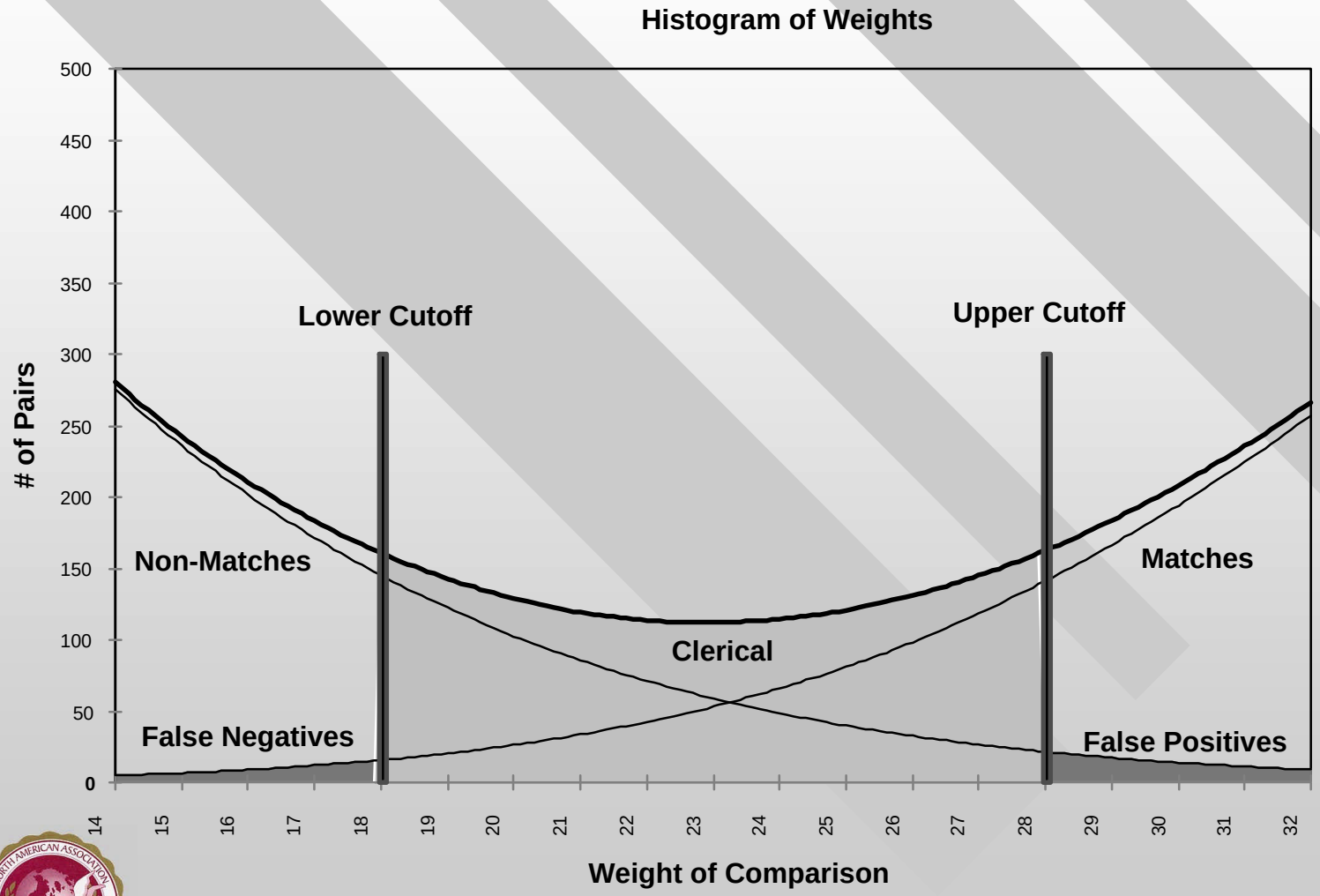
- **It is the difference in the distribution of the weights for unmatched and matched pairs that enables one to discriminate between matches and non-matches.**
- **The more fields are available for matching, the bigger this difference will be and more reliable matches will result.**
- **(so USE all available fields, no matter their condition**

Distribution of Weights



Source: MatchWare Technologies, Inc., Burtonsville, MD, USA (1995)

Detail of Histogram



Source: MatchWare Technologies, Inc., Burtonsville, MD, USA (1995)

Integrity Data
'Re-engineering' Environment

Denver, October 2002

Integrity Data Re-engineering Environment

Ascential Software

- **History**
  - **1980's UniMatch**
  - **1990's AutoMatch (Matchware)**
  - **1998 (?) SuperMatch / Integrity (Vality)**
  - **2002 Integrity (Ascential)**

Brains behind the code: Matt Jaro

August 2002 – Matt Retires, and is livin' LARGE

Integrity Data Re-engineering Environment

Ascential Software

- **Ascential Software**
  - **DataStage - Migration/extraction environment**
  - **MetaRecon – Profiles existing databases (front end for DataStage**
  - **Integrity –Linkage software**
- **My INITIAL impressions:**
  - **Front end wrapper to same ol programs (includes cygwin unix utilities for Win32 (Cygnus software)**
  - **Functionality – basically same as old stuff**
  - **Compatibility – no way to incorporate old AutoMatch setup files into Ingegrity automatically**
  - **Performance – stay tuned...**

Integrity Data Re-engineering Environment

Ascential Software

- **Components:**
  - **Investigation – GUI front end to view Frequency analysis results**
  - **Conditioning – new name for ‘Standardization’**
  - **Matching**
  - **Survivorship – new component used for combining matching (and duplicate) records into the one record that ‘survives’ the match**

Integrity Data Re-engineering Environment

Ascential Software

- **Year 1:**
  - **$10,000 per Cancer Registry Center (either Unix or Intel)**
  - **Includes: License, maintenance, support, 1 seat in Ascential training facility 4 day course, and access to E'Learning tools**
- **Future years:**
  - **$6,000 per Cancer Registry Center**
  - **Includes: License, maintenence, support, and access to E'Learning tools**
- **Additional Seat in 4 day course: $2,995**
- **Onsite Consulting - $7,500:**
  - **4 day includes analysis of datasets used for linkage and basic training and assistance with tuning applications**

Integrity Data Re-engineering Environment
Ascential Software

- **Next version (4.1) of Integrity will have the Clerical Review module reinstated. Estimated release date: 2/03**
- **NAACCR 2003 Meeting – Joint Integrity Training ???**
- **Kimberly Siegel**
  Director of Consulting Practices
  Office 617 210 0842
  Cell 617 470 4731
  kim.siegel@ascentialsoftware.com

- **Also feel free to contact Rich for status update of Software**