

Beider-Morse Phonetic Matching: An Alternative to Soundex with Fewer False Hits

by Alexander Beider & Stephen P. Morse

This article appeared in Avotaynu: the International Review of Jewish Genealogy (Summer 20008).

Background

Searching for names in large databases containing spelling variations has always been a problem. A solution to the problem was proposed by Robert Russell in 1912 when he patented the first soundex system. A variation of Russell's work, called the American Soundex Code, was used by the Census Bureau to facilitate name searches in the census.

Simply put, soundex is an encoding of a name such that names that sound the same will get the same encoding. A search application based on soundex will look for matches of the soundex code rather than matches of the name itself, thereby finding all names that sound like the name being sought.

As an example, the American Soundex code for Schwarzenegger is S625. If the name was misspelled as Shwarzenegger, the code would still be S625, so any search application based on American Soundex would still find the match in spite of that misspelling. However if the name was misspelled as Schwartsenegger, the American Soundex code would be S632, so a search application based on American Soundex would not find the match with that misspelling.

A major improvement to soundex occurred in 1985 with the development of Daitch Mokotoff (DM) Soundex by Randy Daitch and Gary Mokotoff. DM Soundex is a soundex system optimized for Eastern European names. Under DM Soundex, the correct spelling, Schwarzenegger, has two codes, namely 474659 and 479465. The incorrect spelling, Shwarzenegger, has the same two codes, and the incorrect spelling, Schwartsenegger, has the DM code of 479465, which is one of the two codes for the correct spelling. So a search application based on DM Soundex would find the match with either of these misspellings. This illustrates the advantage of DM Soundex over American Soundex for Eastern European names (Austrian in this case).

Both of these soundex systems have, nevertheless, a major disadvantage – they generate many false hits, requiring the researcher to wade through a lot of extraneous matches. The phonetic-matching method proposed in this paper attempts to alleviate that situation.

Main Principles

Beider-Morse Phonetic Matching (BMPM) was developed by Alexander Beider (Paris) and Stephen P. Morse (San Francisco). Beider dealt with the linguistic part of this method and Morse with the computer aspects and all technical issues. Major algorithmic decisions are due to common efforts of both authors.¹

The main objective of BMPM consists in recognizing that two words written in a different way actually can be phonetically equivalent, that is, they both can sound alike. But unlike soundex methods, the “sounds-alike” test is based not only on the spelling, but on linguistic properties of various languages.

For common nouns, adjectives, adverbs and verbs this task is of limited interest. Except for orthographic and typographic errors, these words rarely have spelling variations. The situation is different for proper nouns (i.e., names) – they can appear in documents written in different languages and spelled according to the phonetic rules of the language of the document. Determining that two different spellings correspond to the same name becomes even more difficult when the two spellings use letters from different alphabets.

As an example, consider the name Schwarz (standard German spelling). It can appear in various documents as Schwartz (alternate German spelling), Shwartz, Shvartz and Shvarts (Anglicized spellings), Szwarc (Polish), Szwartz (blended German-Polish), Șvarț (Romanian), Svarc (Hungarian), Chvarts (French), Chvartz (blended French-German), Шварц (modern Russian), Шварцъ (Russian before 1918), שװארץ and שורץ (Hebrew), and שוואַרץ (Yiddish).

In its current implementation, BMPM' is primarily concerned with matching surnames of Ashkenazic Jews. This is due to the list of languages whose graphic and phonetic features are already taken into account. These languages are Russian written in Cyrillic letters, Russian transliterated into English letters, Polish, German, Romanian, Hungarian, Hebrew written in Hebrew letters, French, Spanish, and English. The name matching is also applicable to non-Jewish surnames from the countries in which those languages are spoken.

However the structure of BMPM is general, and we are already planning to extend it to additional languages such as Lithuanian and Latvian. We also plan to incorporate Italian, Greek and Turkish, since this would allow BMPM to be applicable to Sephardic names (as well to non-Jewish names from those countries). In order to extend it to a new language, all we need to do is include of supplementary rules specific to that language. The rules are not hard-coded into the program; instead the phonetic engine is table driven and all that is necessary is to add additional tables to support the additional languages. A description of the different tables involved is presented below.

BMPM is designed to be used as a programming tool, and an individual would be very hard-pressed to do the calculations manually. To use the system, a user would enter a name on a form, that name would be transmitted to a server running the phonetic engine that would generate the BMPM code, and that code would then be compared to the BMPM codes that were previously generated for all the names in a specific database. The steps of this comparison are described in the following sections.

Step 1. Identifying the Language

The spelling of a name can include some letters or letter combinations that allowing the language to be determined. Some examples are:

"tsch", final "mann" and "witz" are specifically German

final and initial "cs" and "zs" are necessarily Hungarian

"cz", "cy", initial "rz" and "wl", final "cki", letters "ś", "ł" and "ż" can be only Polish

More often, several languages can be responsible for a letter or a letter combination. For example, "ö" and "ü" can be either German or Hungarian, final "ck" can be either German or English, "sz" can be either Polish or Hungarian. Sometimes it can be easier to name the language or the languages in which the letters in question can never occur. For example, "y" and "k" are not present in Romanian, "v" can not be Polish, the string "kie" can be neither French, nor Spanish.

The current version of BMPM includes about 200 rules for determining the language. Some of them are general whereas other include the context in which they are applicable (e.g., beginning or the end of a word, following or preceding some letters). The processing of these rules yields one or several languages that could, in principle, be responsible for the spelling entered by the user.

One option of the BMPM engine allows for specifying the language explicitly. That would apply when the database is known to be in a specific language, in which case each name in that database can be encoded using the rules of that language, and the language-determination test need not be done.

Step 2: Calculating the Exact Phonetic Value

In a number of languages, forms of surnames used by women are different from those used by men. For example, it would be Jan Suchy but Maria Sucha. And the wife of Mr. Novikov would be called Mrs. Novikova. This occurs in Slavic tongues (including Polish and Russian), Lithuanian and Latvian. Since the name under analysis can, in principle, be feminine, this step starts with replacing feminine endings with the masculine ones.

After the name has been defeminized, the phonetic engine tries to identify the *exact phonetic value* of all letters of the name, and transcribe them into a phonetic alphabet. Since in principle the number of different sounds is huge, we decided to restrict the phonetic alphabet used in BMPM to those sounds that are shared by the languages we were interested in. For example, the difference between Polish "y" and "i" was deliberately ignored because there is no way to express it in non-Slavic languages. Also ignored was the difference between two sounds expressed in German by "ch", those present in words "ach" and "ich". For the same reasons, numerous vowels found in French and English do not figure in our version of the phonetic alphabet, but instead were replaced with closest equivalents found in Germanic and Slavic languages. The retained list appears in the table below.

	Example		Example
--	---------	--	---------

a	Like in <i>part</i>	b	Like in <i>boy</i>
d	Like in <i>dog</i>	e	Like in <i>set</i>
f	Like in <i>flag</i>	g	Like in <i>dog</i>
h	Like in <i>hand</i>	i	Like in <i>Nice</i> (the city), or <i>ee</i> as in <i>fleet</i>
j	Like y in <i>yes</i> , equivalent to German <i>j</i>	k	Like in <i>king</i>
l	Like in <i>lamp</i>	m	Like in <i>man</i>
n	Like in <i>neck</i>	o	Like in <i>port</i>
p	Like in <i>pot</i>	r	Like in <i>ring</i>
s	Like in <i>star</i>	t	Like in <i>tent</i>
u	Like in <i>flu</i> , or <i>oo</i> in <i>good</i>	v	Like in <i>vase</i>
w	Like in <i>wax</i>	x	Like <i>ch</i> in <i>loch</i> ; equivalent to German <i>ch</i>
z	Like in <i>zoo</i>	S	Like <i>s</i> in <i>sure</i> , or <i>sh</i> in <i>shop</i>
		Z	Like <i>z</i> in <i>azure</i> ; equivalent to French <i>j</i>

Generally, the signs for sounds conventionally chosen by us are the same as those used by International Phonetic Alphabet (IPA). The only exceptions are S and Z, whose IPA's equivalents are ʃ and ʒ, respectively. Our choice was dictated by limiting ourselves to standard Latin characters present on any keyboard using the Roman alphabet.

The transcription of the name into the characters found in the above table (a better term for it would be *mapping*) depends of the result of Step 1. Either Step 1 determined a unique language, or it determined a set of possible languages.

If only one possible language was left after Step 1 the phonetic engine transcribes the spelling to the phonetic alphabet using rules specific to that language. In BMPM, every language possesses its own set of rules for this mapping (less than 40 for Romanian, about 80 for German and more than 130 for Polish). For example, if the language is German, then some of the rules are

"sch" maps into the "S" of our phonetic alphabet

"s" at the start of the word and "s" present between two vowels becomes "z"

"w" becomes "v"

For certain languages, some letters can be read in several ways. In these cases, the phonetic engine assigns them two (or more) elements from the phonetic alphabet. For example, Polish "a" normally corresponds to phonetic "a". In some cases, however, this letter can result from Polish "ą" in which the diacritic sign (comma under the "a") was lost. In this example, the phonetic value would be either "om" (before "b" or "p") or "on" (before other consonants).

If Step 1 resulted in more than one possible language, the phonetic engine processes the name using generic rules. To adequately support the languages of the current version of BMPM, we needed to write more than 300 generic rules. There are two types of such generic rules – ones that are language independent and ones that apply only to certain languages.

An example of a language-independent generic rule is the rule for final "tz" – it can be pronounced only as English "ts". Such language-independent generic rules are applied regardless of which languages are present in the output of Step 1. Other generic rules might be applicable, however, to specific languages only. The output of Step 1 would determine whether or not these language-specific generic rules would be applied. For example, "ch" can be mapped (using the signs of our conventional phonetic alphabet) to "x" in Polish or German, "S" in French, or the diphthong "tS" in English or Spanish. If during Step 1 we learn that English, Spanish and French are not possible, only the Polish/German language-specific rule will be applied, causing the "ch" to be mapped to "x".

Once the name is processed by either the generic rules or the language-specific rules, the phonetic engine applies to the resulting string of phonetic characters a series of phonetic rules that are common to many languages. As an example, consider the rule known in linguistic literature as *final devoicing*. It applies to many European languages, such as German, several Slavic tongues including Russian and Polish, and some dialects of Yiddish. Final devoicing states that at the end of the word the voiced consonants are pronounced as their unvoiced counterparts – i.e., "b" is pronounced as "p"; "v" as "f"; "d" as "t" etc. The phonetic engine takes this peculiarity of speech into account and keeps in the final position only the unvoiced consonants. For example, Perlov gives Perlof. Another rule, also applied by the phonetic engine, is that of *regressive assimilation*, whereby a consonant acquires characteristics of the consonant that follows it:

Voiced consonants become unvoiced when followed by unvoiced consonants.
For example, "b" before "s" is pronounced as "p": Shabse is equivalent to Shapse

Unvoiced consonants become voiced when followed by voiced consonants.
For example, "t" before "z" is pronounced as "d": Vitzon becomes Vidzon

At the end of Step 2 the initial surname is transformed by the phonetic engine into one or several strings of characters that we call the *exact phonetic value*.

Step 3: Calculating the Approximate Phonetic Value

After the rules mentioned in Step 2 are applied, the phonetic engine applies a series of additional rules. These rules take into account the fact that some sounds can be interchangeable in some specific contexts that are more complex than the contexts considered in Step 2 ("beginning/end of word" or "previous/next letter"). For example, in Russian and Belarusian unstressed "o" is pronounced as "a". As a result, Mostov and Mastov sound alike because the first syllable is unstressed. On the other hand, there is no interchangeability in the stressed position: Kats and Kots sound differently. Since automatic determination of the stress position is non-trivial, we decided to deal with "a" and "o" as *approximately* interchangeable. Other rules allow for phonetic proximity of a pair of sounds resulting in their partial confusion. For example, "n" before "b" sounds close to "m" and Grinberg becomes *approximately* equivalent to Grimberg. (Note that in Spanish this equivalence is total. Consequently, in Argentina Grinberg and Grimberg are *exactly* equivalent.)

Just as in Step 2, the *approximate* rules applied here can be either language-specific or generic, depending of the results of Step 1. To adequately handle the languages of the current version of BMPM we needed to write about 200 rules common to all languages, about 120 generic rules (some of which are limited to certain languages), and several dozens language-specific rules per language.

At the end of this step the initial surname is transformed by the phonetic engine into one or several strings of characters that we call the *approximate phonetic value*.

Step 4: Calculating the Hebrew Phonetic Value

All previous steps, even if they were primarily designed to process Ashkenazic Jewish surnames, can in principle be applied to other cultures too. This step, on the other hand, is specifically Jewish. The main aim of this step consists in taking into account the fact that the initial name as written in Latin or Cyrillic characters can be the result of a transliteration from Hebrew. Such spellings are commonplace in various materials related to the Holocaust. Numerous memorial (yizkor) books of communities from Eastern Europe are written in Hebrew and, as a result, the names they mention appear in Hebrew characters. Many lists from these books were transliterated by Jewish genealogists, and in many cases the resulting spellings using Latin characters are simply educated guesses. In the online searchable database of the Holocaust victims provided by Yad Vashem in Jerusalem, many surnames from interwar Poland fall in this category – they appear on the pages of testimony compiled in Hebrew during 1950s and 1960s, and the spelling using Latin characters often represents a guess by Yad Vashem's employees.

Since some vowels do not appear in Hebrew spelling and the sounds of other vowels and certain consonants are ambiguous, a transliteration of the same name from Hebrew to Latin characters made by different people can yield different results. For example, פסטר can yield Fester, Faster, Paster, Pastar, Pester, Fasater, Psater etc., בין can correspond to surnames that were spelled in German as Bien, Bin, Bühn, Bün and Bein, פרימס can be Frimes or Primas.

This step is designed to fix the issues related to the transliteration from Hebrew. To accomplish this, the phonetic engine takes the results of Step 2 and applies a series of additional rules that allow for the ambiguity of certain sounds when dealing with the Hebrew spelling. At the end of this step, the initial surname is transformed by the phonetic engine into one or several conventional strings of phonetic characters that we call the *Hebrew phonetic value*. Surnames whose Hebrew spelling is the same have the identical *Hebrew phonetic value*. Some examples are Bader and Beder; Brak, Berak and Barak; Bober, Buber and Bubar; Brauner, Bronner and Bruner; Mandel and Mendel; Thaler and Teller; Zipper and Ziffer.

Note that the *Hebrew phonetic value* calculated here can apply to surnames that are spelled in Latin, Cyrillic or Hebrew characters. In all these cases, the original characters have already been mapped into the characters of the phonetic alphabet during Step 2. As a consequence, this step deals with strings of phonetic characters only.

Step 5: Searching for Matches

Applications of name matching involve searching for names in electronic lists. Some examples of lists that are of interest to us are:

Names mentioned in reference books on Ashkenazic surnames by Alexander Beider and Lars Menk, all published by Avotaynu Inc. (1993-2008)

Names present in sources related to the Holocaust such as the Yad Vashem list of names, necrologies from various memorial (yizkor) books, lists of inhabitants of various ghettos, prisoners of concentration camps such as Dachau etc.

Names appearing in Ellis Island Passenger Lists

Names extracted from the Polish or Russian civil records and indexed by the JRI-Poland project

Names used by Jews in Argentina

The phonetic values (*exact*, *approximate*, *Hebrew*) of the name being searched for needs to be generated by the phonetic engine at the time the search is performed. But prior to doing any searches, the phonetic value of each of the names in the list needs to be calculated. Some simplifications can be used when processing the entire list of names because there might be information known about the language and the spellings used within the list.

For example, in reference books on Galician and German Jewish surnames, the orthography of all names conforms to the German spelling. As a result, during Steps 2 and 3 every name is processed by the set of rules specific to the German language. The case of Jewish names from Argentina is more ambiguous: some names are spelled in Spanish, others in German, Romanian or Polish. But even in this situation, the processing is simplified because we know that such languages as Hungarian, French or English are irrelevant and, as a result, numerous rules used during Steps 2 and 3 (those restricted to these languages) can be ignored.

The matching of individual name to names present in specific electronic lists proceeds in the following way:

If the one of the *exact phonetic values* of this name and a name from the list are identical, we say that the match is *exact*. These two names are phonetically equivalent.

If one of the *approximate phonetic values* of this name and a name from the list are identical, we say that the match is *approximate*. These two names can be (or not be) phonetically equivalent.

If one of the *Hebrew phonetic values* of this name and a name from the list are identical, we say that the match is *Hebrew*. These two names can be phonetically equivalent only if at least one of them was originally spelled in Hebrew. If the user knows that neither of them was spelled in Hebrew or results from the transliteration from Hebrew, the *Hebrew match* is of no importance and can be simply ignored.

Matches done by BMPM are not necessary *commutative*, i.e. if a surname A matches a surname B, this does not imply that the surname B will match the surname A. For example, the list of surnames present in "A Dictionary of Jewish Surnames from the Kingdom of Poland" contains the names Bak and Bāk: if a user searches for the name Bak, he will get Bāk among the approximate matches, but if he searches for the name Bāk he will not find Bak.

The absence of commutativity occurs because the phonetic engine processes the name entered by the user different from the way it processes the names in the list – in the former case the engine allows for the possibility that some of the diacritical marks (e.g., the mark under the “a”) were omitted by the user, whereas in the latter case the engine assumes that all names in the list have been proofread and are known to contain all necessary diacriticals. So the name Bak entered by the user could also be Bāk, but Bak appearing in the list is really Bak and never Bāk.

Implementation Issues

The result generated by the steps above is a set of one or more sequences of phonetic characters. However computers are much more efficient at matching numerical values from some small space than in matching arbitrary character strings. For this reason, the following additional steps are performed on the phonetic values before matching is attempted:

Each phonetic character is assigned a digit so that a sequence of phonetic characters can be replaced by a numeric value. This numeric value can be quite large, depending on the number of phonetic sounds in the name being encoded.

The resulting number is reduced to a small number space by taking it modulo some base value. This has the disadvantage that two names that are unrelated phonetically can wind up with the same numeric value. Although this is possible, the likelihood of it happening is small, especially if the base value is carefully chosen. For example, that number should not be a multiple of ten, because then only the trailing phonetic characters would be represented and the leading ones would have no effect on the result.

It should be noted that all the sounds in the name contribute to the BMPM phonetic value, and subsequently to the resulting numeric value. This is in contrast to soundex methods in which (1) some sounds such as vowels do not contribute and (2) the latter letters in a name have no bearing on the resulting code value since the codes truncate after four consonants in American Soundex and six in Daich Mokotoff Soundex.

Comparison to Daitch-Mokotoff Soundex

Soundex is one of the solutions proposed in the past to solve the problems of name matching. It has several variants of which the Daitch-Mokotoff (DM) method is the one that is the most commonly used in the domain of Jewish Ashkenazic genealogy.

When soundexing, any letter either receives a numerical value, or is simply omitted. Different consonants can receive the same numerical values, for example, b and v, m and n, g and k. All vowels are treated as interchangeable. As a result, contrary to BMPM, soundexing does not search for the equivalence of sounds: even different (but sometimes close) sounds can match. Consequently, when matching names, soundexing may have a significantly larger number of false positives than BMPM. On the other hand, it can find some true matches that are not found by BMPM because the equivalence is not purely phonetic.

The domain in which soundex seems to be more appropriate than BMPM is when the original form of the name (which is the form as it appears in the list) is not known and all that is known is the form of the name used today. Here are some examples:

Various names starting with Silver – such as Silverberg, Silverstein. Here, Silver came from the original German Silber (or Yiddish "zilber"). But the change is not just phonetic, it is partly semantic – the German/Yiddish word for "silver" is replaced with its English equivalent

Names having English "stone" instead of German "stein" (Yiddish "shteyn") – such as Rotstone instead of Rotstein. The DM value for both of them is the same, though the pronunciation of these two words is significantly different. (The situation is different in the case of "green" for "grün" and "field" for "feld": they do match in BMPM too because here the match is phonetic as well).

Tartatski/Tartatzki/Tartacki becoming Tartaski in US. Here we are dealing with anglicizing – the consonantal cluster "tsk" never occurs in English whereas "sk" is commonly used. Again, phonetically speaking, Tartaski and Tartaski are not equivalent and for that reason BMPM does not consider them as matches.

In the examples above, DM Soundex can find some Anglicized fits for the following reasons:

Adaptation of sounds from one language to another often changes them to sounds that are different, but still close (and consequently their DM-code can be identical)

English is a Germanic language, that is, from the same linguistic group as German and Yiddish. That means that semantic adaptations of Ashkenazic surnames (like Silber to Silver) can produce forms that are close both phonetically and semantically.

DM-Soundex codes include only six digits. So forms shortened by immigrants to a name that contains less than seven consonants (or consonant clusters) can match under DM Soundex. BMPM values are based on the entire name, no matter how long it is. For example, both Konstantinovsky and Constantine have the same DM Soundex code but not the same BMPM values.

On the other hand, here are some cases for which neither DM Soundex, nor BMPM will find matches:

Numerous names ending in *ovsky/ovski/owski* for which their ending were Anglicized to *osky/oski*

All translations to words sounding different such as Schwarz to Black, and Adler to Eagle

All shortened forms that include more than six consonants.

Hebraicized names will rarely give matches by DM-Soundex because Hebrew is a Semitic language, not from the same family as German/Yiddish/Slavic languages. Moreover, often the Hebraicizing involves some shortening and/or change of letters, which will present problems for BMPM as well. Examples are Perski to Peres, Rabichev to Rabin, Scheinerman to Sharon, Gryn to Ben Gurion, Meyerson to Meir, Shertok to Sharett, Shkolnik to Eshkol, Brog to Barak, not to mention Ezernitsky [Jeziernicki] to Shamir, and Mileykovsky [Milejkowski] to Netaniahu.

Summarizing the above, DM Soundex is more appropriate than BMPM for individual searches made by descendants of immigrants to North America or England who know the names of their ancestors in their Anglicized form only. In that case the disadvantage of the large number of false positives is outweighed by the advantage of finding some Anglicized forms that would otherwise not be found. DM Soundex is also more appropriate in cases when a matching should be done between two lists of names, one of which deals with original name and the other with the Anglicized versions. For example, someone may be searching for matches between names in the Ellis Island passenger records (which contain the original European names) and the US census records (in which names have already been anglicized).

In other contexts, BMPM is more appropriate than DM. These include:

Automatic processing by computer of large data bases in order to find matches between elements of various data bases. This was the primary objective that led to the conception of BMPM. If DM Soundex were used in this context, the computer would not be able to weed out the large number of false positives that would be generated.

Searching for individual original names (names used before immigration and not yet anglicized) in large databases. If we want to quickly find matches between two spellings both of which correspond to the European forms, BMPM will immediately provide the list of fits. In this case, the main advantage of DM (finding of some Anglicized forms) is irrelevant. As a result,

if someone knows roughly what the original name of interests was, BMPM will be much more appropriate because it will immediately cover the identicalness of numerous variant spellings of Schwartz (given at the beginning of this article), without polluting the list by the presence of numerous false positives.

There is also a group of matches found by BMPM that are not found by the current version of DM Soundex. Below are several examples, along with the reason why they do match in BMPM:

Triphthongs are *approximately* equivalent to diphthongs: Altmayr matches to Altmayer, Heyem to Heyem, Kajm to Kaiem

Forms with "h" between vowels or at the beginning of the word are *approximately* equivalent to those in which "h" was lost: Johanes and Joanes, Halperin and Alperin

The letter combinations "inm" and "jnm" are *approximately* equivalent to "im" and "jm": Weinman(n) and Weiman(n), Fajnman and Fajman

"sc" before a vowel is not equivalent to "s" or "sch", it can be *exactly* equivalent to "sk": Boscowitz and Boskowitz, Muscat and Muskat

When one sound expressed in our conventional phonetic alphabet by the signs "S" (English "sh"), "Z" (French "j"), "s" and "z" is followed by another sound from the same group, it can be dropped (due to the phenomenon of the *regressive assimilation*, discussed above in this article). As a result, the following names match *exactly*: Hirschstein and Hirstein, Ovruchsky and Ovrutsky

The sound "d" disappears if it is followed by the sound "t" or a diphthong that starts with "t" (such as that expressed by "ch" as in English "check"). Consequently the following match *exactly*: Gladtko and Glatcke, Goldzweig and Golzweig, Kurlandchik and Kurlanchik

Several transliterations into English of Cyrillic vowels followed by "e" are *exactly* equivalent: "ae", "aye", "aie" and "aje" [all for Cyrillic "ae"]; "oe", "oye", "oie" and "oje" (all for Cyrillic "oe") etc. Examples: Faer, Fajer, Faier and Fayer (Cyrillic Фаер), Meer, Mejer, Meier and Meyer (Cyrillic Мееп). In D-M Soundex the forms with "ae", "oe", "ee" do not match to "aye-aie-aje", "oye-oie-oje", "eye-eie-eje", respectively.

Initial "Rh" is *exactly* equivalent to "R": Rhau and Rau, Rhein and Rain

Evidently, some of these drawbacks of the DM-Soundex can be easily eliminated by introducing new rules (for example, the last one). For others, the logic of the DM-Soundex prevents such pairs from matching.

The above arguments show that globally speaking BMPM and DM are complementary tools: each of them has contexts in which its application is more appropriate than that of another method.

¹ The initial work on this algorithm was based on the article by Alexander Beider “Some Issues in Ashkenazic Name Searches” (*Avotaynu: The International Review of Jewish Genealogy*. Vol. XXIII, Number 1, Spring 2007, pp.3–13) and the long term desire of Stephen P. Morse to ameliorate the engine of his various online searchable databases (<http://stevemorse.org>) including Ellis Island Passenger Lists. The initiation of this project (and, more precisely, the personal meeting of its two authors in Newark in July 2007 and their decision to work together) was made possible due to the organisational efforts by Sallyann Amdur Sack and the sponsoring provided by the International Institute for Jewish Genealogy (Jerusalem). The two authors would also like to thank Logan Kleinwaks, Gary Mokotoff and Jean-Pierre Stroweis, who tested the draft versions of BMPM and provided numerous valuable comments.