Move similarity analysis in chess programs

D. Dailey, A. Hair, M. Watkins

Abstract

In June 2011, the International Computer Games Association (ICGA) disqualified Vasik Rajlich and his Rybka chess program for plagiarism and breaking their rules on originality in their events from 2006-10. One primary basis for this came from a painstaking code comparison, using the source code of Fruit and the object code of Rybka, which found the selection of evaluation features in the programs to be almost the same, much more than expected by chance.

In his brief defense, Rajlich indicated his opinion that move similarity testing was a superior method of detecting misappropriated entries. Later commentary by both Rajlich and his defenders reiterated the same, and indeed the ICGA Rules themselves specify move similarity as an example reason for why the tournament director would have warrant to request a source code examination.

We report on data obtained from move-similarity testing. The principal dataset here consists of over 8000 positions and nearly 100 independent engines. We comment on such issues as: the robustness of the methods (upon modifying the experimental conditions), whether strong engines tend to play more similarly than weak ones, and the observed Fruit/Rybka move-similarity data.

1. History and background on derivative programs in computer chess

Computer chess has seen a number of derivative programs over the years. One of the first was the incident in the 1989 World Microcomputer Chess Championship (WMCCC), in which Quickstep was disqualified due to the program being "a copy of the program Mephisto Almeria" in all important areas. See *ICCA Journal* 12/4. The conclusion here was rather clear, and left little room for contention; later cases would not always be so easy.

The emergence of open-source software being run on general-purpose hardware, in particular Crafty in the mid 1990s, led to another prickly situation, with the 1996 WMCCC seeing Gunda participate, though it was derivative of Crafty, who was also in the field. It should be noted that the Gunda authors fully acknowledged the debt to Crafty, and there were side issues with Gunda being from the host institution. See the editorial in *ICCA Journal* 20/1.

The 2003 World Computer Chess Championship (WCCC) saw the competitor List disqualified due to a problem with timely inspection of program code (following a request by the tournament director). The author was later exonerated from any specific wrongdoing, upon code inspection (at a later date) by an

outside examiner. See [14] and also *ICCA Journal* 28/1. A successor to List, namely Loop in the 2007 WCCC, is currently under investigation by the ICGA as a possible Fruit derivative.

Although never disqualified, the ElChinito author Eugenio Castillo Jimenez admitted re-using Crafty source code in an open letter dated Aug 2004 [16]. The WCCC entrant named Chinito in 2002 and 2003 was never directly linked, and Jimenez participated in later ICGA events, with XieXie in Chinese Chess.

The 2006 WCCC saw another disqualification, with the LION++ team being dismissed for not duly informing the tournament committee that their program was derivative of the open-source Fruit (which had competed in 2005, and would permissibly provide the basis for cluster-based entrants in 2007 and 2008). They had acknowledged Fruit in their program notes, but not as required on the entry form, and not received the necessary permission from Fruit's author Fabien Letouzey in any case. Two inspectors concluded that LION++ was a "close derivative" of Fruit, and thus it was disqualified. See *ICCA Journal* 29/2.

Another example is Squarknll being refused admission to the 2010 WCCC on the grounds that it was a barely-modified version of the open-source program RobboLito (itself of rather dubious pedigree). See [18] for more.

In other competitive games, such as computer go, one can give the example of Silver Igo, alternatively named KCC Igo (or KCC Paduk), and commercialised as Ginsei Igo, which was claimed to derive from Handtalk [29], written by Chen Zhixing. This was reported on in 2000 by Keene [17]. Zhixing has given various evidence for this [27], and also concerning a program named Hamlet, alternatively called Hit, AI-Igo 7, or Medusa (see [28]). Due to unresolved issues from a decade previous, KCC Paduk was denied entry to the 2008 International Computer Games Championship [5].

1.1. Other computer chess derivatives

When expanded to programs that do not compete in live tournaments, the list of derivatives is probably too long to enumerate, though Ron Murawski has a partial list at [19]. We should mention here in particular two examples involving Rybka, both of which have some bearing on the events recounted below. The first was the release of Strelka, initially in mid-2007, and then later in a source code version in early 2008. While initially being rather permissive toward Strelka, the Rybka author Vasik Rajilch eventually declared Strelka 2.0 to be a virtual clone of Rybka 1.0, and this seems to be adequately substantiated via outside examination and general consensus.[1]

Secondly, in 2009 an open-source program called IPPOLIT (later morphed into RobboLito and IvanHoe) was released, and Rajlich indicated that it was a reverse-engineered clone of Rybka 3. Subsequent analysis by Watkins (see [24]) couched this view more cautiously, saying that undoubtedly the makers of IPPOLIT had reverse-engineered Rybka 3 as a starting point, but that vari-

[1] Ciancarini and Favini mention Rybka/Strelka in [CF], but with no definitive conclusion.

ous differences (and simplifications) existed so as to place it beyond the label of merely a "clone" as the term would be used in its most strict sense.

1.2. Outline of this paper

We first very briefly recall the history of Fruit and Rybka with the ICGA decision. In particular, we highlight the call for more "objective" methods to assess program similarity. In response to this, we take up the work of Dailey, involving a similarity-detection scheme that operates by determining whether two programs make overly similar move-selections. Though not suitable for all purposes, this similarity detection mechanism seems adequate to meet the desired criterion of objectivity.

We then present the data accumulated by Hair using the Dailey detector, and enlarge upon the analysis he gave concerning it. The specific Fruit/Rybka numbers are given. We do not view them as definitive, but rather supplementary to the ICGA investigation, and in particular they indicate that there was sufficient cause for suspicion.

We then discuss some alternative methods of move similarity analysis, and turn to questions involving the robustness of the Dailey tester, such as whether varying the time alloted has much of an impact on the results, and whether strong engines tend to have larger intra-group move similarity than when weaker engines are included. We also note that move similarity data may be of interest beyond questions of clone detection, particularly to questions of "style" of play.

Finally, we conclude with some brief complementary comments on creativity and originality in the field of computer chess.

2. History of Fruit and Rybka

In March 2011, the International Computer Games Association opened an investigation concerning the entries of Vasik Rajlich into ICGA tournaments, particularly the World Computer Chess Championship (WCCC). This was predicated on a formal complaint by Fabien Letouzey, the author of Fruit, who averred that certain versions of Rajilch's program Rybka were derivative of Fruit, and thus did not meet the ICGA originality rule. Letouzey himself had competed in the 2005 WCCC with Fruit (finishing 2nd), and then indirectly in 2007 and 2008 when Fruit formed the basis for GridChess and ClusterToga.

After a month or two of sorting through evidence, the Secretariat of an investigation panel submitted a consultative report on the matter to the ICGA Board (in the person of David Levy, its president). Rajlich was invited to join the panel in this evidentiary phase, but chose not to do so. Levy then requested that Rajlich present a defense. This was largely met with a non-response, with Rajlich first querying what rule he was claimed to have broken, and then merely citing CCRL `ponderhit` data (see §4.3.1). The ICGA Board concluded unanimously that Rajlich had broken the WCCC rules, and chose to disqualify all his entries from 2006-10 (including WCCC victories from 2007-10), and furthermore banned him for life from participating in ICGA events. See [15] and *ICGA Journal* 34/2 for more information.

The primary technical evidence given in the case was a comparison of the evaluation functions of Fruit and Rybka, the former from source code, and the latter by disassembly. Rybka, at least in versions 1.0 to 2.3.2a (from Dec 2005 to Jun 2007), was found to have an abnormally large overlap with Fruit in its selection of evaluation features, and this was considered sufficient to breach the ICGA's originality requirement.[2] One complaint by Rybka defenders was that this evaluation feature comparison was "subjective", which was inherent to the methodology used.[3]

2.1. Later responses

In July 2011, Rajlich gave an interview with Nelson Hernandez [12]. In it, he again opined for more objective measures, such as move similarity detection. In January 2012, ChessBase published a four-part series in defense of Rajlich from Søren Riis. The most relevant part of his work appears in pages 12-14 of [22]. We make some analysis of this in §4.3 below.[4]

3. Move similarity detection

A tool to assess move similarity in chess engines has been developed by Dailey [6]. It consists of a Tool Command Language (Tcl) program that contains 8239 chess positions, which are sent to the engine that is being tested via the UCI protocol (or Winboard, if a suitable adapter is used). The engine is sent the command "`go depth 50`", and then after a desired interval, the `stop` command is sent. One initial intent of Dailey was to keep the search time as small as reasonably possible, as the intent was to judge evaluation rather than search, the former being more domain-specific to chess. Some engines can be finicky, such as crashing in certain positions, using too much time in pre-evaluation (as they expect long searches), ignoring the `stop` command, etc. However, in general one achieves good results.[5] Below we comment on variations of the

[2] Some other considerations also played a role in the decision. For instance, Rajlich entered Rybka versions into private tournaments in 2004, and these were found to have re-used significant amounts of Crafty code [26, §3], including copying of hash bugs and obsolete tablebase workarounds; and the ICGA investigation further noted a general lack of continuity between these 2004 Rybka versions and Rybka 1.0 (Dec 2005), with the latter containing numerous (unexplained) bits that bore atypical resemblance to Fruit, in both the principal engine parts (search and eval) and also in support routines and data structures [26, §4].

[3] The comparison, which followed works of Wegner and Watkins, was eludicated by the latter via 40 pages of analysis and discussion, but subjectivity must ultimately play a role.

[4] Rajlich, in conjunction with Chris Whittington, subsequently filed a complaint against the ICGA with the FIDE Ethics Commission (see [8], case 2/2012), and this is still pending.

[5] A side issue, which could become more relevant if/when the Dailey tester is used more widely, is how robust the Dailey detector is to attempts to beat the system. A first issue, that the 8239 positions are available, can be overcome simply by having the testers change them (see §6.2.3). Similarly, the ruse of having an engine act differently for analysis and game-play (for instance, differentiating "`go infinite`" from "`go wtime`") should be detectable with a bit of effort. However, the effect of (say) perturbing every `evaluate()` call by a small random amount is something that could be studied.

test (see §6), such as using longer search times, or using depth-limited searches rather than time-limited.

3.1. Adjusting for engine strength

While one can indeed instrument the Dailey detector by simply giving all engines the same amount of time to search each of the 8239 positions, an additional idea (in concordance with above desire to accentuate evaluation aspects) is to scale the amount of time alloted (per position) to each engine by its perceived strength.[6] From the standpoint of engine originality, this timing adjustment will tend to eliminate various aspects of a merely technical nature, such as making an executable faster (via a better compiler, or by more arduous means such as refashioning data structures). However, making an engine stronger is indeed a valid goal in itself, thus making it unclear whether one should apply any such "correction" during the data collection phase

In the end, our main dataset (derived from that of Hair) used the time scaling explained in the next paragraph, while in §6.2.2 we make some comments about how the observed data differ when a uniform time allotment is given. In §6.2.1 we additionally discuss what happens when *all* engines have their time allotment increased, say by a factor of 2 or 10.

The specific Elo adjustment used in Hair's original experiment (see §4.1) was based on various tests he had carried out previously. For instance, in [10], he presents data to suggest that currently almost 120 Elo can be expected from doubling the thinking time, this being derived from experiments with multiple engines at time controls from 6s+100ms up to 96s+1.6s.[7] From this, he chose a formula of $2^{(\mathrm{Elodiff})/120}$ for his strength scaling, with Houdini 1.5 the baseline given 20ms, and approximate ratings obtained from various sources, such as those listed in Footnote 6.

3.2. On the philosophy of the move similarity paradigm

One problem with move similarity detectors and their "objective" limit is that it can often lead to attempts to beat the system, such as (say) now occurs with haemoglobin testing in Olympic events.[8] Indeed, in their work on clone detection, Ciancarini and Favini [CF] propose having the automated systems be a first screening, with then humans making the final decision. Depending upon the competence and capacity for work of such umpires, it would seem best to draw the line for initial suspicion rather low.

---

[6] An estimate of the latter can usually be determined various rating lists, such as CCRL or CEGT, or from ChessWar or WBEC data.

[7] In a later thread [11], he gives data from 1m+1s up to 8m+8s for one specific engine, with doublings giving about 80 Elo each. In [LN, page 192] one finds an estimate of 50-70 Elo when doubling the thinking time. A different estimate can be derived from [HN]: *the curve shows that a fourteen-ply search yields an additional 125 rating points over a thirteen-ply search*; with the effective branching factor of today's engines near 2.0, this reasonably matches Hair's data, and 13-14 ply searches currently take about a second to perform.

[8] For instance, the Austrian skiers/biathletes in Turin were careful to be under the "objective" limit, but the IOC still acted when complementary evidence of doping was obtained.

Another issue is that, particularly when search is modified and evaluation is kept the same, two programs can have highly similar move selections for positions where static features are dominant, but differ when dynamism is of more import. Here we can note that the ICGA currently has an investigation open concerning Thinker in the 2010 WCCC, with the claim that its evaluation derives from Strelka and/or IPPOLIT (and the move similarity confirms this), while at the same time Thinker is often named in Internet fora as producing some of the most interesting move selections among second-tier engines. One possible explanation is that the "positions of interest" to Internet commenters are often of a more dynamic nature than the typical position.

Finally, there is precedent for abnormally high move similarity using perfectly legitimate techniques. One early example is the Deep Thought team, who tuned their evaluation function to moves from grandmaster games [HACN]. Expert-driven genetic algorithms (see [DKN]) have also been observed to increase move similarity, with an increase of coincidence to the mentor's moves being permissibly obtained. Automatic learning of evaluation, such as fitting moves to a dataset as in [B], could also lead to greater coincidence. Related to this, Hair comments (see [7]):

> *The use of statistical methods assumes that the authors have access to a common pool of ideas, but that there are no interactions between authors/engines. In reality, authors/engines do interact.*
>
> *This tool should not be used solely for determining derivatives and clones. Other methods should be used in conjunction with this tool. Ultimately, any accusation of cloning requires an examination of the code of the accused author.*

4. Analysis of similarity data

The genesis of our work came about due to some data collection and analysis done by Adam Hair. We enlarge on this work here, recapitulating the methods, and indeed subsume the relevant data collection in our download package [7].

4.1. The initial dataset of Hair

The main dataset considered below was collected by Hair on a 3.05 Ghz QX6700 computer running Windows XP 64-bit Professional operating system. Some of the other data collected (by Hair) comes from a 2.95 Ghz Intel E8400. These are approximately equal in computation speed, based upon a Crafty 19.7 benchmark from Bryan Hoffmann.[9] All engines were tested in single-cpu mode (one core), with a hash size of 64MB.

The lowest time interval that was used was 20ms. There are two issues with this: first, a global one, that time slicing of the operating system might intervene;

[9] The experimental effects of such timing variations are discussed in §6.2.1 and§6.2.2, while Footnote 37 comments about calibration via such benchmarking.

and second, an engine-specific one, in that many engines only poll for input every 10ms or so. The experiments in §6.2.1 and §6.2.2 indicate that overall these should not matter too much, and the results we observe are typically within the experimental precision one expects from the size of the data pool.

Hair took 378 total versions of engines, involving 99 (presumably) independent families, and instrumented them in the Dailey test. He then performed an analysis of the data, restricting to one engine per family. The purpose of this procedure, with its family-based restriction, was to attempt to measure what the "expected" move similarity should be for engines that are unrelated. The resulting distribution of move similarity can then be analysed, for comparison with externally observed data (as in §4.2.3).

4.1.1. Possible data anomalies

This initial dataset of Hair was collected in more of an observational manner than a scientific one; for instance, Hair did not always ensure that every engine responded to the tester commands correctly, and was sometimes unable to determine which engines should be classified in the same family. In §4.2.2 below, we attempt to "correct" his initial data by filtering out what appear to be data anomalies.[10] These include some undetected relatives (which would subvert the independence assumption) and also a few engines that have significantly lower move-matching than the average. Furthermore, in §6.2 we note some data from redoing the experiment in a more stringent manner.

4.2. Expected move similarity for "unrelated" engines

The conditions listed in §3 were used, with the strongest engine (Houdini 1.5) receiving 20ms, and the others scaled according to formula given in §3.1. One of the 8239 positions is missing; perhaps Hair removed it because it caused some engines to crash. It is difficult to estimate the experimental precision, though in a statistical sense one expects (cf. coin-flipping) that a normal distribution with $\sigma \approx \sqrt{8238}/2 \approx 45$ should ensue, implying a 95% confidence level of about ± 90 moves matched.[11]

The results from 8238 positions were then compared pairwise upon restricting to one engine in each of the 99 families, yielding $\binom{99}{2} = 4851$ data points for counts of moves matched. The data show a semi-reasonable fit (see Figure 1) to a normal distribution with a mean of 3720 moves matched (45.16%) and a standard deviation of 235 moves (2.86%).

[10] Note that we filtered on a basis specifically derived only from the *observed data*, and not on (say) anything of a more anecdotal nature that Hair might tell us. Similarly, although we later asked Watkins about possible Fruit relatives (see Footnote 12), we filtered these purely based upon their abnormally high move-matching.

[11] One can validate this by (say) a bootstrapping technique – take sequences of 8238 positions *with replacement* from the Dailey set, and analyse the resulting move-matching distribution.

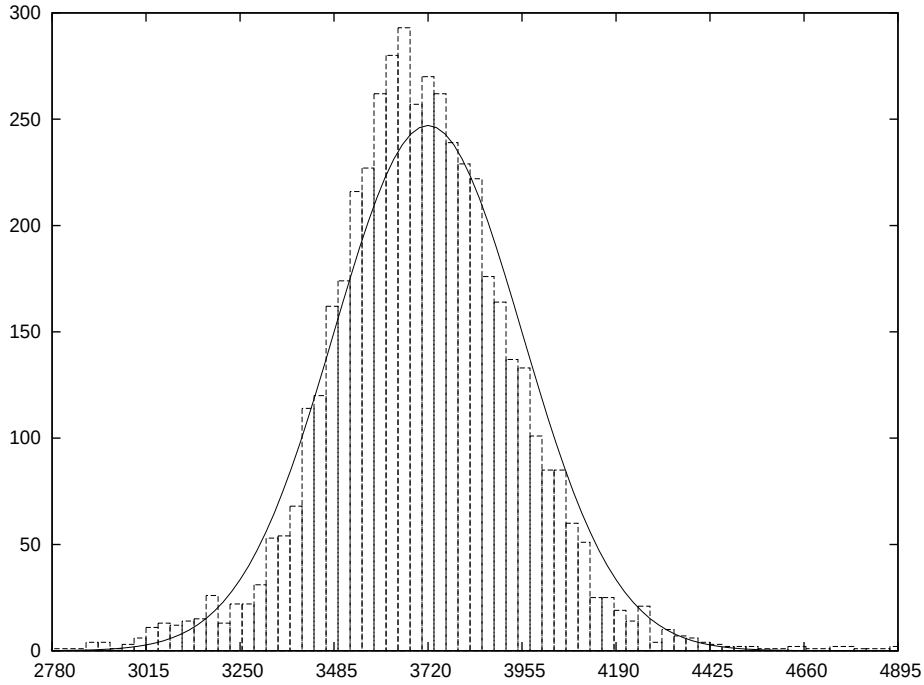

Figure 1: Distribution of 4851 engine pairs for move-matching counts (main Hair dataset)

4.2.1. Non-normality of the data

We can make a few comments about the non-normality of the data. The tails are longer than a normal distribution (the excess kurtosis is 2.24), particularly at the end corresponding to larger move similarity. The latter could be due to various undetected derivatives/relatives being included.[12] The skewness is 0.263, and might be indicative of general sharing of ideas of engine authors. Outlying pairs at the end of smaller move similarity, particularly when one engine occurs repeatedly in such pairs,[13] suggest something amiss with the testing protocol, such as not searching to the desired time limit for all positions.

Consistent with the observation of long tails, the results are somewhat softened if we use L-moments instead. The coefficient of L-variation is 0.035 (compared to 0.063), while the L-skewness is 0.019, and the L-kurtosis is 0.170.

[12]For example, it appears that (at least) the four most extreme points listed in Hair's README (Onno/Fruit, and Houdini/Critter/Robbolito), are generally accepted to have shared ancestry (at least in the versions given). Furthermore, the other extreme points include: Naum/Thinker, with the versions used both now thought to derive from Strelka; Philou/Stockfish, the former thought to be derivative of the latter; and various pairwise comparisons with Hamsters/Rotor/Colossus/Alaric/Onno/Naraku/Fruit, with it possible these are all Fruit-related to varying degrees (see Footnote 17 for more).

[13]The most notable engine here is Quazar, and Junior is not too far behind. GNU Chess also had sufficiently low move-matching (both average and maximal) that we chose to err on the side of caution and omit it also.

4.2.2. Filtration of anomalous data

We chose to retain the entire Hair data set, even with the above caveats. However, we should also relate the following. Upon reducing the above listed suspected relatives (in Footnote 12) to include but one engine per grouping (Alaric, Houdini, Naum, and Philou), and also eliminating GNU Chess, Junior, and Quazar (Footnote 13), the mean varies only slightly to 3723, while the standard deviation decreases considerably to 185. The skewness increases to 0.315, while the excess kurtosis becomes slightly negative at -0.102. We give a graph of the resulting distribution in Figure 2.

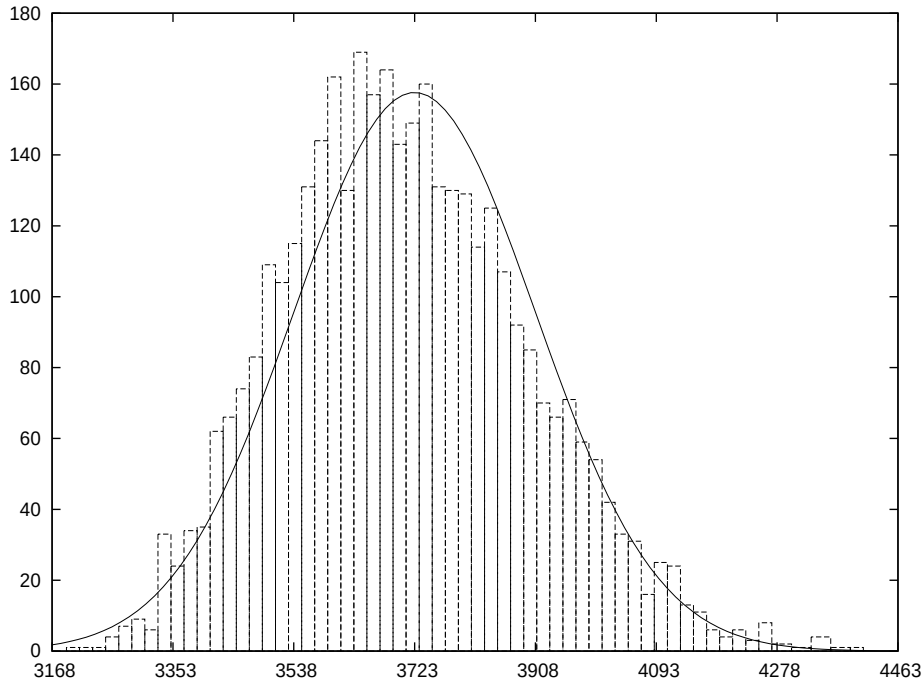

Figure 2: Distribution of 3655 engine pairs, upon filtering for perceived data anomalies

In Table 1 we give an explicit list of filtered engines and the reason why. An asterisk by the name indicates that it was *not* filtered, rather was kept as the representative of said family.

In Table 2 we give the top 30 move-matching numbers from Hair's 99 representatives. Only Cheese/Glass (in 29th place) would remain upon filtration. Note that Rybka 4.1 (not Rybka 1.0) is the "representative" for this family in Hair's data, so the data point of Fruit 2.1 and Rybka 1.0 does not appear (it is 4593, and would thus be in 13th place). Similarly with Rybka 2.3.2a (which matches Fruit 2.1 for 4672 moves).

engine	reason
Robbolito	Early Houdini versions match exact node counts at low depth
Critter	Source code perusers opine it is a Robbolito "hybrid"
Naum*	Disassembly found Strelka/Rybka evaluation re-used
Thinker	Disassembly found Strelka/Rybka evaluation re-used
Philou	Disassembly found Stockfish evaluation largely re-used
Fruit	Open-source engine that is frequently reused
Alaric*	Disassembly found Fruit PST and other elements
Colossus	Disassembly found Fruit mobility and PST elements
Hamsters	Quick disassembly found Fruit-like PST used
Naraku	Disassembly found Fruit evaluation largely re-used
Onno	Essentially admitted to be a Fruit re-implementation
Rotor	Large Fruit move-matching, inconclusive quick disassembly
GNU Chess	Unclear if was responding correctly to tester
Junior	Data seem notably spoiled, likely problems with tester
Quazar	Data seem notably spoiled, likely problems with tester

Table 1: Thirteen filtered engines from Hair's list of 99 representatives

4.2.3. Adopted standards

Based upon a version of the above work of Hair, the Dutch computer chess organization (CSVN) has chosen to adopt a standard (based on the Dailey test) for admission [3]: *When a percentage of 60% or more is found in the similarity test the tournament organisation will not accept the entry of the chess engine in the tournament.*[14] The CSVN appends a clause that allows them to overrule their 60% limit. However, in §7 we argue that this limit already expresses something close to near-certainty, rather than just reasonable suspicion.

4.2.4. Fruit and Rybka

The move matching with Fruit 2.1 and Rybka 1.0 Beta (Dec 2005) was 4593 of the 8238 positions (55.8%), while the move matching with Fruit 2.1 and Rybka 2.3.2a (Jun 2007, so contemporaneous with the 2007 WCCC) was 4672 (or 56.7%). Both of these appear to be rather significant statistical events.[15] There are various difficulties with a specific quantification, due to the non-normality as noted in §4.2.1 and also the (quite notable) decrease in standard deviation upon "correcting" possible data anomalies. Furthermore, as noted in the Introduction, we view this result as merely supplementary to the ICGA investigation, and indicative of sufficient cause for suspicion.[16]

[14] Added in proof: The CSVN is now using a protocol designed by Richard Pijl and Marcel van Kervinck (see [20]). It uses similar principles to the Dailey test, but (for now) seems to omit any specific numbers for when an engine would be disqualified or investigated further.

[15] Note that Rybka 4.1 is the representative of the Rybka family in the 99 engine sample of above, so neither data point reported here would have been used in analysing that distribution.

[16] Regan has similarly opined (for human move-matching) that a 1 in 1000 match in conjunction with "external" evidence is stronger overall than a statistical 1 in a million match by itself.

Fruit 2.1	Onno 1.04	5499	Alaric 707	Rotor 0.6	4541
Houdini 1.5	RobboLito	5061	Onno 1.04	Hamsters	4527
Critter 1.2	Houdini 1.5	5018	Fruit 2.1	Naum 4.2	4524
Critter 1.2	RobboLito	4942	Rotor 0.6	Hamsters	4492
Alaric 707	Fruit 2.1	4925	Fruit 2.1	Rotor 0.6	4477
Philou 3.60	Stockfish 2.11	4833	Onno 1.04	Thinker 5.4d	4468
Naum 4.2	Thinker 5.4d	4817	Naum 4.2	Onno 1.04	4465
Alaric 707	Onno 1.04	4786	N2 0.4	Rotor 0.6	4459
Colossus	Fruit 2.1	4771	Fruit 2.1	Philou 3.60	4421
Colossus	Onno 1.04	4694	GarboChess	Onno 1.04	4420
Naraku 1.31	Onno 1.04	4668	Alaric 707	Hamsters	4418
Fruit 2.1	Naraku 1.31	4658	Fruit 2.1	Pupsi2	4413
Colossus	Naraku 1.31	4591	RobboLito	Rybka 4.1	4409
Onno 1.04	Rotor 0.6	4567	Cheese 1.3	Glass 1.8	4401
Fruit 2.1	Hamsters	4544	Fruit 2.1	Thinker 5.4d	4399

Table 2: Top 30 move-matching data in Hair's 99 representative engines

In later sections we attempt to put this result in more context. For instance, we: vary the amount of time given to the engines (§6.2.1); remove the strength-based scaling (§6.2.2); restrict the data to exclude some of weaker engines (§5), which may be making more suboptimal moves; and compare to results from other commercial engines from the 2005-2008 period (§8).

In all cases, it appears that the Rybka/Fruit move similarity is still rather notable, though the quantification of this can vary.

4.2.5. Different statistical measurements

The above tries to create an "expected" distribution of move similarity from suitably independent engines, and then use this to analyse data external to this. An alternative comparison scheme is simply to count the number of data points in the 4851 engine-pair sample that exceed the external data. For instance, there are 10 pairs that exceed the 4672 move-matches of Fruit 2.1 and Rybka 2.3.2a, and 2 more that exceed the 4593 of Fruit 2.1 and Rybka 1.0 Beta. However, as noted in Footnote 12, it is perhaps not unreasonable to label all 12 of these data points as suspicious (some more than others), and upon removing all the relevant engines from consideration, one might prefer to regard (say) the 4401 move-matches between Cheese and Glass as the maximum among unrelated engines.[17] This would then be considerably less than the observed data with Fruit 2.1 and the relevant Rybka versions.

[17]Decisions on what to include in the pool of "unrelated" engines remain one of the more difficult aspects in designing a testing system. Presumably one can determine a statistical test in this regard, but regarding Footnote 12: Naraku 1.4 is known to be IPPOLIT-based and Naraku 1.31 is thought to be from Fruit; disassembly (by Watkins) of Alaric and Colossus showed enough evaluation similarities (scoring and otherwise) with Fruit 2.1 to be classified as Fruit-inspired; Hamsters and Rotor were less conclusive in his brief inspection. *(continued)*

One can also measure the expected move similarity between engines of the *same* family (cf. the question raised near the end of §4.3). We give a couple of relevant examples of this. There are 18 (explicit) Fruit variants in Hair's data (including 8 Toga variants and GambitFruit, with these not being directly from Letouzey), thus giving us 153 pairs. These give move-matching counts ranging from 3597 to 7232, with 44 of the 153 counts less than 4593 (this being the Rybka10/Fruit21 count). Similarly, there are 11 Rybka variants catalogued,[18] giving move-matching counts ranging from 4332 to 7179, with 25 of the 55 pairs less than 4593.

4.3. Other methods of discerning move similarity

In [22, page 13], Riis displays a dendrogram produced by Kai Laskos.[19,20] The method of analysis for this is not explicitly given, with Riis stating that it "confirmed what was already known" and that it shows "that the similarity between Fruit and Rybka 1.0 is actually not that large, and certainly not when compared to other programs."

It is not clear to us how Riis comes to the conclusion quoted above. For instance, in his text,[21] Riis compares the red circle at distance 10 (where Rybka 1.0 diverged from Fruit 2.1) to data at distances 1-3 for some quite strongly related clones – but he does not consider whether distance 10 itself is already enough to be overly related (for ICGA purposes or otherwise). Note that Shredder 10 diverges from Shredder 11 at this same distance, and Rybka 3 diverges from Rybka 4 even further away at distance 11.

Indeed, Riis quotes Laskos as noting that distance 20 is where the relationships are most likely statistical noise, and both of the other data points (in the dendrogram of Laskos) that are below this distance have at least some relation. Namely, at distance 15 we find Komodo/Rybka3 which share Larry

---

Another possible family reduction could be Rybka/Houdini, though we chose to retain this, as Rybka4/Houdini15 is under the Cheese/Glass limit. Similarly with Thinker/Rybka (and possibly Fruit) – the Thinker version here has Strelka's evaluation, and thus that of Rybka1, with the relation of the latter to Fruit being the point in dispute.

[18] It should be said that the Rybka data do tend to split up into 2 parts, the first corresponding to Rybka 1 through 2.3.2a, and the second for Rybka 3 and later versions, subsequent to Kaufman's work on the evaluation function. In the same vein, Fruit 1.0 is not much like other Fruit versions, and one can also note that the versions for each of these sets were released rather irregularly, sometimes with many changes, other times with comparatively few.

[19] Riis says Laskos: *performed a statistical analysis of ponder-hits using a similarity testing program written by chess programmer Don Dailey*; where it seems Riis has conflated the technical term "ponderhits" (as Rajlich had mentioned, see §4.3.1) with the Dailey detector (which is what Laskos actually used).

[20] The specific TalkChess post where Laskos gives the dendrogram (that Riis displays) is in the Engine Origins subforum, which is accesible to members only. Laskos has produced a number (25 or so) of these dendrograms which appear in various forum threads, and the characteristics can vary depending upon the pool of engines used in the comparison.

[21] Riis slightly errs with his comment that "The brown circle is where IvanHoe 47c diverted from Rybka 3", as this brown circle at distance 3 is actually where the given versions of Houdini, Strelka, Critter, and IvanHoe diverge, whereas the Rybka3/IvanHoe divergence is at distance 6.

Kaufman as the author of the evaluation function; while at distance 16 there is Stockfish/Fruit, where both of these are open-source.[22]

Riis then mentions additional analysis from Miguel Ballicora to bolster the contention that the Fruit/Rybka relationship is not very significant.[23] Riis concludes by stating: "[w]ith more data you soon realize that the distance between Rybka 1.0 Beta and Fruit 2.1 is *not* less than other unrelated engines" – however, even accepting this conclusion (and our comments in §4.2.5 raise doubts), it again fails to inquire whether the distance *is* less than (or similar to) that between known related engines, as in the examples given above. In other words: among engine pairs at (or near) the calculated similarity distance of Rybka/Fruit, what is the percentage of unrelated engine pairs? It does not seems that this question has been addressed (either by Riis or in TalkChess posts), and it seems critical toward an understanding of dendrological methods.

One possible difference between the experiment of Laskos and that of Hair[24] is that the latter adjusted the amount of the time given to engines based upon their strength, while this is unclear for the former.[25] We consider the effect of this in §6.2.2 below.

4.3.1. `ponderhit` *data*

If one reads the statements of Rajlich closely, he actually refers not to Dailey's similarity tester, but to the `ponderhit` data of the Computer Chess Rating Lists (CCRL), whose methodology is given in [2].[26] Various questions can be raised therein. For instance, regarding the ignoring of drawn games,

[22] Tord Romstad, the author of Stockfish precursor Glaurung, was rather well-acquaintanced with Letouzey and indeed was selected by Letouzey to post the open letter about Rybka online.

[23] Again the argument looks to be made across sundry TalkChess posts. Hair indicates that he sent data to Ballicora (who is a biochemist), who then processed them with something that looks similar to PHYLIP (a computational phylogenetics package). This uses neighbor-joining to form a consensus tree.

It appears that Ballicora makes two points. His first is that the Fruit/Rybka relationship is not significant because none of the consensus trees put them in the same branch. It is not clear what "branch" means here, and this could depend on whether engines closely related to Fruit 2.1 (see Footnote 12 for a partial list) or Rybka 1.0 (such as Naum/Thinker/Strelka) are included. Hair indicates he has more recently sent Ballicora some data from a filtered set of engines, and Rybka/Fruit were then indeed identified in the same branch – so it seems the question is thereby reduced to couching the proper sense of what "filtration" to use.

Ballicora's second point, which is less technical in nature, appears to be that the Rybka/Fruit move similarity data by itself (however it is quantified) is not sufficiently flagrant to warrant such a definitive decision as made by the ICGA. This concords with the comments made in §3.2, and indeed, the ICGA Rules indicate abnormal move similarity as a precursor to further investigation.

[24] Hair similarly produces a dendrogram for some his data (included in [7]), but he chooses to restrict to one engine per family, and uses Rybka 4.1 rather than Rybka 1.0 in the chart. In any event, one would still need to determine what "close" should mean in such an analysis.

[25] It seems that Laskos used the constant 100ms default for many engines, but with the dendrogram displayed by Riis, he indicates that: *[f]or the new super-engines included here, I adjusted time/move to adjust for their superior strength.*

[26] Note that `ponderhit` numbers appear only with the data for 40 moves in 40 minutes, which is a dramatically different time allotment compared to that used by Hair (or Laskos).

Rybka was dominant for many years, and so had a lower draw rate than other engines.[27] Related to this is the question of whether the style (intentional or otherwise) of an engine can cause a tendency toward positions with larger or smaller best/next-best increment (cf. §4.4 below) – indeed, a similar consideration in human play was the primary impetus of the "complexity" measure given by Guid and Bratko in [GB, §2.3]

Another issue is that some engines are (much) more speculative concerning when to make instant moves, and such moves are ignored by `ponderhit`. Finally, for the statement: *[p]onder hit statistics is much more interesting in middlegame positions, where the move choice actually shows engine playing style and understanding*; this seems reasonable (if debatable), though the choice of instrumenting this limitation via a specific ±9 pawn cut-off again could be given some justification. We were able to find various numbers used for similar purposes, all of which were significantly lower this.[28]

We could not find any analysis of `ponderhit` data, and Rajlich does not specifically indicate how one might conclude that Rybka/Fruit is not anomalous.

4.4. Comparisons to previous move matching data

Note that the percentages for move similarity that are produced by the Dailey program are not immediately comparable to those determined by other means. For instance, the move-matching of Crafty with past world champions given in [GB, §2.4, §3.5, Figure 8], should be seen as making a different measurement. In response to a comment by a reviewer, we attempt to codify this more fully. One principal distinction is that positions in the Dailey test set typically have a smaller difference in evaluation between the best and next-best moves.[29]

We obtained the WCC Dataset [9] from Guid's website,[30] but unfortunately this does not easily specify which positions were excluded (due to being outside the ±2 pawn range). This consists of positions taken from classical World Chess Championship matches, with the purpose being to see which humans matched Crafty's selections most frequently. Due to the difference in purpose, their criteria for excluding positions was not the same as that (presumably) used with the Dailey set.

[27] This removal of drawn games is already of sufficient significance that it should be justified statistically, if it is to be used. One could also ask, e.g., if the loser is more likely to match the winner (or vice-versa) in decisive games, as in the relevant data sample, Rybka would be the winner much more often than the loser.

[28] For instance, Hartmann [H, §7.3] derives that 1.5 pawns is the threshold for a decisive advantage in his evaluation scheme. Similarly, Guid and Bratko [GB, §2.1] use 2 pawns in their human-computer comparisons, though this limit comes about via consideration of when a human might decide to play in a more risk-free manner, while [RMM] uses 3 pawns in a similar context. The Houdini manual [13, §1.1] indicates that Houdini 3 tries to map evaluation scores to self-play winning percentage at blitz time controls, with 3 pawns already being equivalent to about 99% *wins* (not merely scoring percentage).

[29] This is not to be confused with the complexity measure they give in their Figures 1 and 3.

[30] We ignored the included Sep 2006 data (Elista) that post-dated the publication of [GB].

Their 68994 FEN records give 58934 distinct positions, and for each we ran a depth 14 search in MultiPV=2 mode in IvanHoe 9.46h,[31] and eliminated the 6581 positions (11.2%) for which the main evaluation was greater than 200 centipawns in absolute value.[32] We then did the same for the 8239 positions[33] in the Dailey test set, finding 7837 (or 95%) that were not eliminated by this criterion.

It should be pointed out that the above filtering (of one-sided positions) does not eliminate positions where there is a much-preferred move, such a trivial re-capture. Indeed, it seems that the "average" best/next-best increment is a poor statistic, as the Guid-Bratko data ends up being rather skewed by positions where there is a clearly best move. The above protocol gave an average differential of 41.8 centipawns (cp) for the Guid-Bratko set, which (when taking engine differences into account) is reasonably close to the 50 cp one gets from eyeballing the right side of their Figure 8. This same protocol yields an average differential of 15.5 cp in the Dailey set.

However, if we further eliminate positions where the best move is (say) at least 200 centipawns better than the second-best, then 2367 more positions are eliminated (4.5%) in the Guid-Bratko set and the average drops noticeably to 19.5 cp, while the Dailey set has only 22 positions eliminated (0.3%) with the average marginally decreasing to 14.7 cp.

The above should suffice to demonstrate that one cannot compare move similarity statistics from alternate data sets without further thought. In particular, it is meaningless to extract (say) the Kramnik-Crafty move-matching percentage from Guid-Bratko and compare it to those obtained in our analyses.

5. Restriction to strong engines

One criticism of the ICGA investigation was that too many weak engines were used in their comparison (of evaluation functions at the source-code or disassembly level). The same could be said for (some of) the data accumulated by Hair. Here we restrict Hair's data to 12 top engines, namely:

> Houdini, Stockfish, Gull, Spark, Spike, Hannibal, Shredder, Zappa, Komodo, Sjeng, Hiarcs, and Booot.

The representative versions (from Hair's 99 families) for all 12 of these have a CCRL 40/40 rating exceeding that of Rybka 1.0 Beta (on comparable hardware), and are generally thought to be sufficiently independent in their creation.[34]

[31] Guid and Bratko used an unspecified (and modified) Crafty version, but the identity of the engine used presumably matters little for the purposes of comparing typical best/next-best increments of the position sets, as long as it is applied uniformly. We first tried to use Stockfish, but had MultiPV problems; its 4 centipawn granularity could also be questioned.

[32] We also ignored absolutely forced moves, and those where the next-best score was mate.

[33] These are not given as FENs, but as move sequences from the game start.

[34] We excluded RobboLito, Critter, and Rybka as being too related to Houdini, while Naum and Thinker derive from Strelka/Rybka, and Onno is Fruit-based.

The resulting data from the 66 engine pairs saw an average of 3733 moves matched (compared to 3720 in the total pool of 99 engine families), with the minimum 3404 (or 41.3%) from Spike/Komodo, and the maximum 4261 (or 51.7%) with Booot and Stockfish.[35] The deviation was only 165 moves, notably smaller than the 235 in §4.2, and also quite a bit less than the 185 from the filtered set of §4.2.2. The skew increased to 0.68, while the excess kurtosis was 0.60. Recall that Fruit 2.1 and Rybka 1.0 Beta had 4593 moves matched, quite significantly exceeding all 66 data points given here (indeed, this is over 5σ from the mean).

6. Robustness upon varying the experiment

Another criticism about the ICGA investigation was that the result might have differed if the comparison method(s) had been varied. We cannot comment directly on this, but will provide some evidence that the Dailey tester is quite robust to variations in the experiment.

In the subsections below, we shall describe various modifications of Hair's experiment that were carried out, such as varying the time allotment, removing the "correction" for engine strength, or changing the set of positions used.

Furthermore, some of Hair's original data collection could be termed more "observational" than scientific. However, given its status in the computer chess community, it still must be considered a primary source. The data obtained here should be as complementary to the original data.

6.1. A concordance metric for experiments

We first need to describe a metric for judging whether a modification to the experiment had much (or any) effect. We do this via concordance.

Given two experimental datasets, we wish to see how much they differ. One approach to doing this is via a pair-comparison metric (a similar idea appears in [GBT]). Specifically, given any pair of engine-pairs A-B and C-D, we want to know whether two datasets agree on the relative comparison. For instance, if one experiment says that the A-B move-matching exceeds that of C-D move-matching, and the second experiment agrees, then they concord on this observation. More formally, given two datasets X and Y of measurements on engines pairs P, we consider the concordance (with $P_1 \neq P_2$) given by

$$Z_{X,Y} = \#\left\{ \{P_1, P_2\} \subset P \;\middle|\; \begin{array}{c} X(P_1) \geq X(P_2) \text{ and } Y(P_1) \geq Y(P_2), \\ \text{or } X(P_2) \geq X(P_1) \text{ and } Y(P_2) \geq Y(P_1) \end{array} \right\} \Big/ \binom{\#P}{2}.$$

Here $X(P_1)$ means the number of move-matches for pair P_1 in the dataset X. One can describe other metrics, such as weighting by an "assuredness" measure based on $|X(P_1) - X(P_2)|$, but the above seems to suffice for our purposes.

[35] In general the open-source Stockfish had higher move similarity than other engines. The only other data points to lie outside 2σ were Stockfish/Gull and Stockfish/Hannibal.

6.1.1. Expected concordance

It is not easy to determine what the expected concordance between two experiments should be. One cannot simply perturb the data for any individual engine (or engine pair) as the effects across the entire engine pool must be considered. Similarly, the appropriate way to carry out something like bootstrapping is not immediately evident. Our best guidance is that the observed concordance from the 100ms experiments with the Dailey positions and the changed positions (§6.2.3) is a reasonable expectation.

6.2. Catalogue of modified experiments

In §§6.2.1-6.2.3 below we describe various modifications made to Hair's original experiment. For each we give the observed concordance from the restricted pool of 25 engines. Fuller details can be found in [7]. For these modified experiments, we used a pool of some of the stronger engines, namely

> Bobcat, Bright, Cheng, Crafty, Cyrano, ET Chess, Frenzee, Gaviota, Gull, Hannibal, Houdini, The King, Movei, Nemo, Protector, Quazar, RedQueen, Rodent, Ruffian, Shredder, SmarThink, Spark, Spike, Stockfish, and Zappa.

See [7] for version numbers and some discussion as to why various engine were excluded. Junior, Sjeng, and Texel seemed to have too many problems with the tester. Gandalf and Tornado were marginal, but we excluded them.

Various other engines showed minor problems when the time window was short. Most notable here was Komodo (which we excluded in the end), primarily because its adjusted time allotment was so small due to its great strength. However, Komodo's results appeared typical when the time was 100ms or more. Some others, such as The King and Zappa, might be considered iffy (at least at the shorter time windows), but we kept them in the concordance comparison.

Fruit and Rybka were omitted, as they are points for comparison. Bobcat, Cyrano, and SmarThink all had high Fruit influence (nearly comparable to that for Rybka 1.0 in some experiments), but we kept them anyway. Their inclusion tends to skew the distribution, but at the level of standard deviations the Fruit/Rybka anomaly still persists. In the same vein, the inclusion of open-source engines such as Crafty and more notably Stockfish evinced a visible increase in the average move-matching.

One can inquire about other modifications, such as increasing the hash size,[36] or using tablebases. These seem to be delving into minutiæ. A more substantial consideration, using fixed-depth searches, is considered in §6.3.

[36] Lukas Cimotti (the caretaker of the Rybka Cluster) noted in a forum post that Rybka shows much higher self-similarity when the UCI command `Clear Hash` is sent between positions. However, this doesn't seem to have much effect on the comparisons with other engines.

The general question of self-similarity (the number of move-matches an engine yields when the experiment is repeated) is a prickly one, and can depend on, say, how engines poll for input, particularly at shorter time intervals.

The combination of results in §6.2.1 and §6.2.2 makes it difficult to conceive that (within reason) changing the hardware involved could matter too much.[37]

6.2.1. Increasing the amount of thinking time

The first variation of experiment carried out was to increase the time allotments by a factor of 2 or 10, so that the reference engine Houdini 1.5 would have 40ms or 200ms. Here we kept the strength-based time adjustment.

6.2.2. Giving all engines an equal time allotment

The Dailey tester has 100ms as its default allotment. We tested the engine pool with this allotment, and also 50ms and 200ms.

6.2.3. Changing the set of test positions

One can also vary the set of positions used in the Dailey tester. We commented briefly on a related matter in §4.4 above. Here we kept the default 100ms allotment in the Dailey tester.

6.2.4. Tabulation of results

In Table 3 we list the results from the six experiments. The average move matching count and the standard deviation[38] are given, as is the maximum move-matching count from the $\binom{25}{2} = 300$ pairs, and finally the move-matching count for Rybka 1.0 with Fruit 2.1 in the last column.

time	avg	std	max	F21/R10
50ms	3698	150	4151	4399
100ms	3758	148	4239	4468
200ms	3846	154	4445	4651
40ms_adj	3950	131	4306	4757
200ms_adj	4185	129	4653	4878
newpos (100ms)	3644	160	4147	4518

Table 3: Results of modified experiments

One can note that engines do tend to agree more as time increases, and that the time adjustment for strength reduces the standard deviation.

6.2.5. Observed concordances

Table 4 lists the observed concordances. Recall that our principal comparison point is NewPos versus 100ms (0.839). We see that increasing the time had less effect than using new positions, while introducing the scaling had more effect.

[37] For instance, Watkins reports [25] that the IvanHoe/Rybka3 speed ratio (in nps) was 31.58 for one processor and 28.13 for another, or a 12% relative hardware differential – we claim that any effects from such are overshadowed by the experimental fluctuations.

[38] As indicated elsewhere, the distributions differ sufficiently from a normal distribution to make one wary, but we still find this to be the most useful statistic.

	50ms	100ms	200ms	40msA	200msA	NewPos
50ms	- - -	0.894	0.861	0.813	0.805	0.841
100ms	0.894	- - -	0.886	0.800	0.792	0.839
200ms	0.861	0.886	- - -	0.802	0.815	0.859
40msA	0.813	0.800	0.802	- - -	0.874	0.783
200msA	0.805	0.792	0.815	0.874	- - -	0.796
NewPos	0.841	0.839	0.859	0.783	0.796	- - -

Table 4: Concordances from various experiments

6.3. Some other experimental directions

We used fixed-time searches largely out of inertia – namely, Dailey's tester uses them as the default, and Hair's work followed the same path. Other computer chess experiments, notably Guid and Bratko [GB], have used fixed depth searches, for reasons such as platform independence. Our first thought was that these were not specifically applicable to our experiments; for instance, engines will vary in how much quiescence search is used (particularly significant for low search depths), and some even use various unspecified accounting for what "depth" means. Still, in conjunction with the original hope that short time allotments in the Dailey test might tend to bring evaluation more into focus, one might attempt to assess similarity via short fixed depth searches. In response to a reviewer, we give an elongated (but by no means complete) commentary on this below.

Additionally, in an attempt to determine various effects of evaluation in move choice, and following a private suggestion of E. G. H. Schröder, we modified Fruit 2.1 to use the Rybka 1.0 Beta numerology in evaluation. This increased its move similarity with Rybka 1.0 Beta from about 56% to 60%, somewhat less than Schröder had guessed. A similar hybrid that was tested was Komodo with PST tables from IvanHoe, and in Dailey's words "the tester was not fooled", that is, the results still identified this hybrid with Komodo, and not IvanHoe.

6.3.1. Fixed depth analysis

One of the reviewers indicated the opinion that:

> *Intuitively, time-limit based move similarity analysis is more associated with the search-techniques aspects of the programs, while fixed depth should be more helpful to detect similarities/differences in the evaluation functions. That is, if A is a derivative program of B in terms of a similar evaluation function (but uses somewhat different search techniques), then both programs are expected to produce similar results at some fixed depth of search. This is perhaps oversimplified, but it would be interesting to see at least some results of the fixed depth analysis...*

We firstly note (though not in complete disagreement with the above) that intuition could come to opposite conclusion, namely that one could propose fixed-depth tends to accentuate *search* differences, in that it measures how

much/little individual engines prune (or extend) the search tree. If engine A does a full-width 8 ply search, compared to engine B (with the same evaluation) which prunes rather drastically in its "8 ply" search, the move-matching might be rather low due to the vastly greater number of alternatives considered by engine A. Of course pruning for top engines must be quite good (else why would they use it?), but pruning is still only a heuristic and indeed is wrong fairly often. Notably, many modern engines leverage pruning on a *statistical* basis, in that it only need be correct often enough for the resultant increase in "depth" to outweigh the cost. This could contrast with older notions of pruning, where the expectation was perhaps more typically that any pruning measures used should be correct nearly all the time (in the sense of not perturbing the α-β tree).

As with many things in computer chess, until this is put to a suitable experimental test (unfortunately outside the immediate scope of this work), it is perhaps akin to picking horses as to which supposition is correct, and we do not claim to have better handicapping skills than the reviewer. We have not yet designed such a test (let alone implemented it), but perhaps one could (for instance) take most of the "search" pruning/extensions out of an open-source engine (thus the evaluation remains constant, while search is modified), and then test it both at fixed depth and fixed time, seeing which yields more concordance in move selection when compared to the original.

Another reason we initially disdained fixed-depth is that the notion of "depth" in a modern pruning engine is anything if well-defined. Indeed, some engines (notably Rybka) subtract off a fixed amount when reporting their depth to the user, perhaps on the grounds that the lower-ply depths are more of a beefed-up quiescence search that anything else. Others might (say) re-search a previous ply with less pruning if the best move fails horribly at a later. Another element is that in the Deep Blue era engines tended to prefer "extensions", while now "reductions" are more vogue.

Despite all such difficulties, in Table 5 we produce a comparison of about 20 top engines using fixed 8 and 10 ply searches (using the 8238 Dailey positions). As a guideline, Houdini takes ∼75ms on average (on our test computer) to accomplish a 10 ply search, most others taking longer (some significantly).

depth	avg	std	max	F21/R10
8 ply	3794	144	4198	4612
10 ply	3980	152	4377	4781

Table 5: Move-matching with fixed-depth searches

The concordance was 0.763 for depth 8 compared to 100ms, and 0.720 for depth 10 compared to the same. Note that these are lower than all items given in Table 4. However, we had to slightly restrict the engines used here for various reasons (such as Winboard engines not properly responding to fixed-depth searches), so a definitive statement should not be inferred from this. The concordance between depth 8 and depth 10 was 0.867, approximately at the level of 50ms versus 200ms, in accord with the idea that 2 additional ply should correspond approximately to a 4x increase in time alloted (for a modern engine).

6.3.2. Comparison to human moves

Another suggestion from a reviewer was to compare computer moves to those played a human. Again this is stretching the immediate scope of this paper, but we assembled positions from World Championship games played by Botvinnik (chosen mainly due to number of games, and breadth of opponents). Upon eliminating early, late, and one-sided positions and those with a clearly-best move, we were left with 3299 positions. We tested top engines both at 100ms fixed time and 40ms scaled. The inter-computer comparisons are in Table 6.

time	avg	std	max	F21/R10
40ms_adj	1852	55	2051	2180
100ms	1770	70	2013	2047

Table 6: Move-matching with Botvinnik positions

Most interesting was that Botvinnik matched **none** of the engines as much as the minima between the computers. Namely the inter-computer minimum was 1712 moves for 40ms adjusted and 1633 moves for 100ms fixed. The maximum for Botvinnik was 1603 moves (with Houdini) in the 100ms experiment, and 1596 (with Quazar) in the 40ms scaled experiment.

Indeed, the "Botvinnik" engine would have been curtailed from our analysis on the grounds that there must be something wrong with the testing protocol. One possible element is that Botvinnik is rated perhaps around 2700-2800, while the engines at said time controls are actually significantly weaker, at best 2500. As a crude test, we increased the time allotment for Houdini to 200ms and 500ms, and this heightened the move-matching to 1622 then 1666.

Again we are unable to answer all questions posed, and certainly there is room for further investigation here.

7. WBEC Premier: a case study in specifying an objective limit

Leo Dijksman runs private WBEC tournaments,[39] and the Premier Division of his latest tournament occurred in early-mid 2012. We will use this as a test case on how to set an objective testing limit. The analysis we give here (which uses a different metric that the analysis in §4.2, for we now take the *maximal* (unrelated) similarity for each engine, rather than work with all engine pairs), will be used to argue that a 60% limit (as chosen by the CSVN) is rather high, at least with the specific Dailey test with the time allotments used by Hair, and something more like 55-56% should already be a cause for suspicion.

[39] He has since stopped running such tournaments. One of the stated reasons was the increasing number of "authors" who did not know how to fix their bugs, even after he spent quite a long time tracking them down in meticulous detail. These tournaments run by Dijksman over the years were quite popular, particularly with authors who had implemented features (such as automated learning) which the standard rating lists did not test.

Similarly it seems that Olivier Deville's ChessWar series has been suspended, again with the increasing number of clone engines being one of the reasons. It is sad that the fallout has affected these men, both of whom have dedicated so much time and effort to provide testing grounds for both amateur and professional engines alike.

Of the 30 prospective entrants for the Premier Division, a couple of them were clearly derivatives, with more than 60% move-matching (in Hair's data) to some other (open-source) engine. There was also a large gap between derivatives and "clean" engines in these percentages. Indeed, after realising a few name changes (ZChess/Pharaon, Doch/Komodo, and Hannibal/TwistedLogic share authorship), the largest move similarity in Hair's data between a WBEC Premier engine and *any* non-related engine was 53.3% for Hannibal with Philou. Many engines had no move similarity above 50%, except for previous versions of the same engine. To exemplify, in Table 7 we give Hair's data for the most-similar 10 engine versions to Spark 1.0 (Bright, like Spark, is written by Allard Siemelink).

Table 7: Spark move similarity

```
--- Spark-1.0 (82 ms) ---
63.67  Spark-0.5 (83ms)
60.95  Spark-0.4 (103ms)
54.79  Spark-0.3 (129ms)
49.39  Philou 3.51 (1208)
49.20  Bright-0.5c (187)
49.05  Bright-0.3d (240)
48.81  Philou 3.60 (686)
48.62  Stockfish 1.51 (88)
48.48  List MP 11.64 (190)
48.45  Strelka R-3-E (42)
```

Table 8: Highest move similarity (WBEC Premier)

entrant	perc	closest engine
Hannibal 1.0	53.3	Philou 3.51
Booot 5.1.0	53.1	Loop MP 12
Crafty 23.4	52.6	List MP 11.64
Stockfish 2.2.2	51.3	Onno 1.04
Pharaon 3.51	51.1	Delfi 4.6
Komodo 3.0	50.4	Rybka 3 Dynamic
Spark 1.0	49.4	Philou 3.51
Baron 2.23	48.6	GambitFruit 4bx
Frenzee 3.5.19	48.6	Delfi 5.4
Jonny 4.00	47.5	Diablo 0.51
Spike 1.4	47.2	Philou 3.51
Zappa Mexico II	47.0	Strelka R-3-E

Table 8 gives a list of WBEC Premier entrants contained in Hair's data (only 12 of the prospective 30 were so contained), with the highest move similarity for each (after predecessors/relatives have been removed), and the engine (not necessarily a WBEC Premier participant) to which this corresponds.

Baron 2.23 was taken from Hair's data rather than Baron 3.30, and Fruit was ignored altogether (WBEC version 2.6d was not in Hair's data; prior versions have been copied so much as to throw off any analysis). Ktulu had too many problems with the tester. Of these entrants, Crafty and Stockfish are open-source. Philou 3.51, appearing multiple times as a "closest engine" has 5270 move-matches (or 64%) with Stockfish 1.51, and was classified in its family.

The above data sample of 12 engines has a mean of 50.0 and a (sample) standard deviation of 2.2. A graphical representation (cumulative) is given in Figure 3, with the first two standard deviations around the mean being shaded.[40]

One can apply the same "maximal similarity" metric to Hair's data, and in Figure 4 we display this[41] for representatives of 86 post-filtration families. Due to difficulties with resolving all the 378 engines into families, we chose to take

[40] A confidence interval, for either the mean or the deviation, from such a small sample might be larger than desired, but below this result is confirmed from the larger 86-engine set.

[41] Note, of course, some move-counts occur twice, once for each engine involved in the pair.

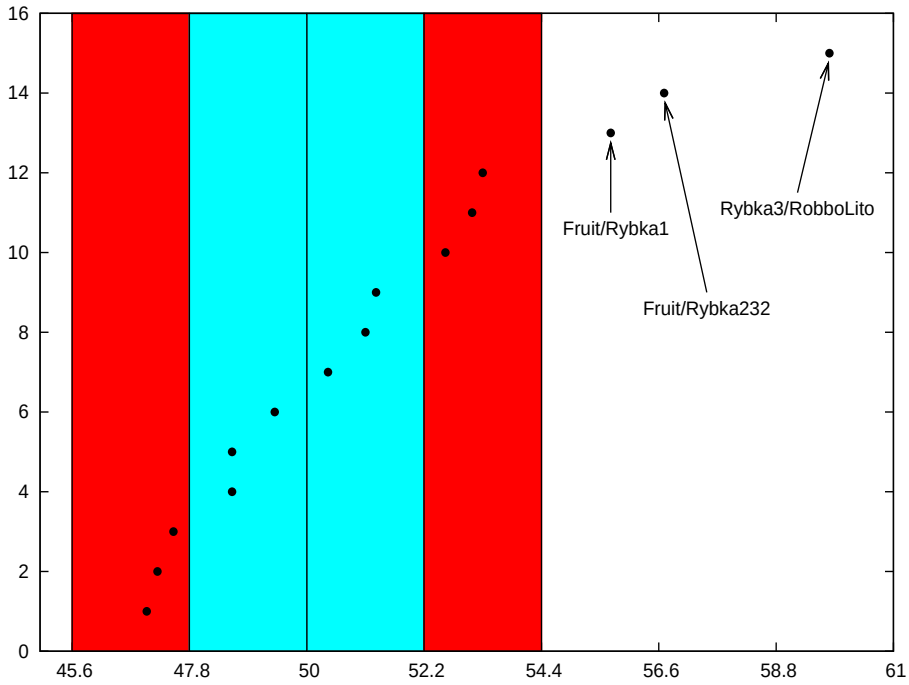

Figure 3: Maximum similarity data (WBEC Premier entrants), with 3 reference points added

the maximum similarity only with respect to these 86 engines. Here the mean is 4117 matched moves (50.0%) and the deviation is 165 moves (2.00%).

Both graphs also depict the data points for Fruit 2.1 with Rybka 1.0 Beta and Rybka 2.3.2a, and also that for Rybka 3 with RobboLito 0.085d1, which at 59.8% (or 4927 moves matched) is still under the 60% limit specified by the CSVN (see §4.2.3). This latter data point is significant since Rajlich himself [21] denoted RobboLito (and kindred) as a Rybka 3 derivative for quite some time (and [24] mostly concurs).

Note that in these graphs we took the *maximum* similarity from Hair's data for each engine, while the previous analysis took *all* similarity observations. It is unclear which method is superior, and could depend on the purpose. The method used here might tend to delimit the influence from weaker engines.

In any event, taking the maximum similarity means that we no longer need to count *pairs* of engines when analysing the likelihood of a rare event. This should make it easier to determine whether an observed move-similarity is indeed unlikely to be a chance event. It seems reasonable that, in a field of (say) 15-20 entrants, a 2.5σ occurrence (or about a 1 in 160 chance, assuming normal distribution – we refrain from a deeper statistical analysis as this section is illustrative in nature), so a similarity at the 55-56% level, at least with the Dailey test at the time limits used by Hair, is already enough to be suspicious, particularly (one might add) when with respect to an open-source engine.

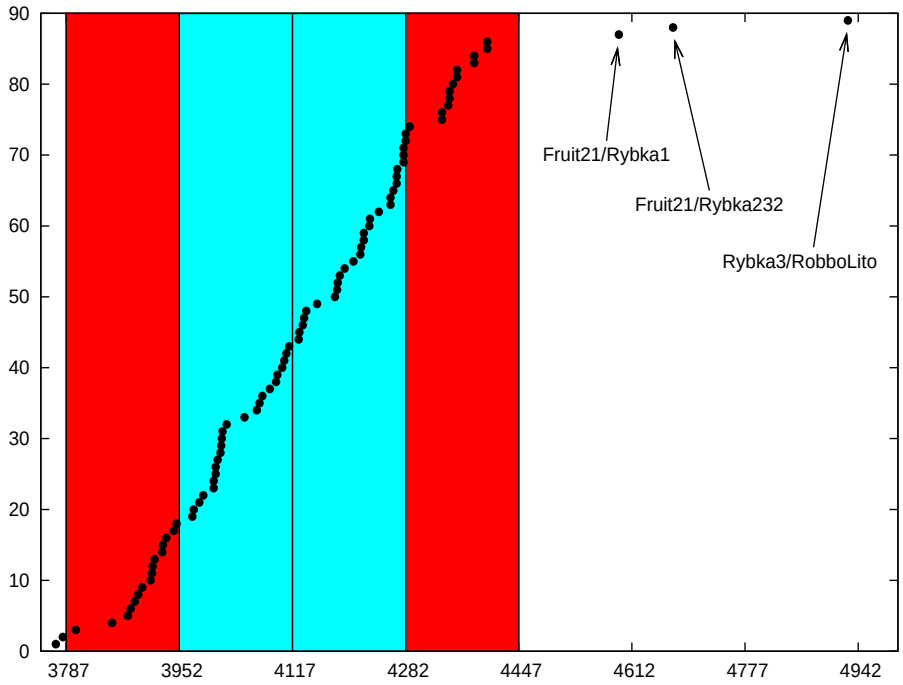

Figure 4: Maximum similarity data from 86 families (filtered), with 3 reference points added

8. Other programs

Yet another claim that has come up in the Fruit/Rybka saga is that other programs (particularly commercial competitors of Rybka) have also gained via re-use of Fruit. For instance, in an interview soon after the Rybka 1.0 Beta release (Dec 2005), Rajlich opined [4, #20]: *Yes, the publication of Fruit 2.1 was huge. Look at how many engines took a massive jump in its wake: Rybka, Hiarcs, Fritz, Zappa, Spike, List, and so on.*

In Hair's data, it appears that Rybka is the only contemporary version of the enumerated programs to show much Fruit influence.[42] For instance, Spike 1.0 (Aug 2005, two months after Fruit 2.1) has only 4008 moves matched (48.7%) with Fruit 2.1, less than the 4076 (or 49.5%) of Spike 0.9a (Apr 2005). Similarly, List 5.12 had 3926 moves matched (47.7%) while the successor LoopList had only 3874 matched (47.0%). Finally, Zappa 1.1 (Aug 2005) has 4082 moves matched (49.6%) with Fruit 2.1, much less than the 4592 of Rybka 1.0 Beta.

[42]The comparative sizes of the "massive jump(s)" made by the listed engines can also be studied: Spike 0.9a is 2419 on the CEGT 40/20 list, while Spike 1.0a Mainz is 2472; Zappa 1.0 is 2379, while Zappa 1.1 is 2412; Fritz 8 Bilbao is 2520 while Fritz 9 is 2586; HIARCS 9 is 2481 while HIARCS 10 is 2572 (cf. the next footnote); List 5.12 is listed in CCRL 40/40 at 2618, while LoopList (Oct 2005) is 2659. There is thin information for Rybka, based only upon its performance in ChessWar events. One can estimate that Rybka 1.6 (from 2004, competing in ChessWar V Section D) was very roughly 500 Elo behind the engines listed above, while Rybka 1.0 (Dec 2005) at 2677 on CEGT 40/20 is ahead of them all. Rajlich himself made a wild guess that Rybka had gained 20 Elo from Fruit influence (see [4, #21]).

Though HIARCS 10 (Dec 2005) was not in the original Hair data, a copy was provided by Mark Uniacke, and using the testing methodology described in §3 the number of move matches was 3836 (or 46.6%).[43]

We were unable to get any numbers for Fritz 8 (Nov 2002, or Aug 2004 for the "Bilbao" version) or Fritz 9 (Oct 2005).[44] It might be noted that later versions of Loop (starting perhaps with version 10.32f from mid-2006) are currently under investigation by the ICGA, with the claim that they are mostly just a transliteration of Fruit.

8.1. Other strong commercial engines in the Hair data

We complete this section by noting some other strong (and commercial) engines in Hair's data. These include HIARCS 12 (March 2008), with maximal move similarity of 3928 positions (47.7%) with "N2 0.4" and Shredder 11 (Oct 2007) at 3944 positions (47.9%) with Strelka R-3-E.[45] These results (which are the *maximal* move similarity for each engine) are all significantly lower than the move similarity between Fruit 2.1 and either Rybka 1.0 Beta or Rybka 2.3.2a.

Indeed, subsequent to Larry Kaufman's work on the evaluation function in Rybka 3, the move similarity between Rybka and Fruit 2.1 dropped considerably, to 52.4% with Rybka 3 (Aug 2008), and 51.0% with Rybka 4 (May 2010).

9. Observations on originality and creativity

The history of computer chess has seen a varying balance between competitive and scientific aspects. In various back issues of *ICCA Newsletter/Journal* (see 4/2 page 17 or 6/4 page 25 for example) one can see the cutthroat tactics that were employed in the early days, particularly when commercial interests were at stake. On the other hand, at least until Deep Blue defeated Kasparov in 1997, there was a good sense of the importance of scientific advancement, with the renaming of the *Newsletter* as the *Journal* in 1983, with consequent enlargement of contributions of more academic interest, being an important milestone.

The aforesaid Deep Blue match (and result) seems to have diminished the scientific impetus in computer chess, and the reasons for this are certainly multiform, though a principal one must be that the triumph of brute-force methods

[43] The surmise that gains in HIARCS came from the Fruit release was roundly rejected by team member Enrico Carrisco [1], while on the HIARCS Elo improvement graph webpage, Uniacke notes [23]: *HIARCS 10 became the biggest strength update in 12 years and jumped to the top of a number of rating lists. The first professional version - I began working full time on HIARCS in 2003 so this is my first full time version with 2 years behind it and a revolutionary new openings book.* One can also note that the gain seen already in HIARCS 9.6 (late 2004) is roughly 55 of the 95 Elo between the 9th and 10th versions.

[44] The best surrogate we can are able to offer for Fritz 9 is the CCRL `ponderhit` data (at 40/40). Here Fritz 9 is listed at 60.4% ponderhits with Fruit 2.2.1 from 4130 positions. To compare, Fruit 2.1 with Rybka 1.0 32-bit is 63.2% from 3431 positions, and Rybka 1.2 with Fruit 2.2.1 is 64.9% from 2009 positions. The predecessor Fritz 8 Bilbao had 61.7% ponderhits with Rybka 1.2 from 1204 positions. See caveats in §4.3.1.

[45] Junior 10.1 (Dec 2006) misbehaves with the tester; 3484 matches (42.3%) is its maximum.

(as compared to something more human-imitative) tended to imply that the field had become more of an engineering endeavour, rather than one where truly novel approaches to tree-search in chess would be seen.[46] The past 15 years have not particularly confirmed this view, as it is readily apparent that at least 400 Elo has been gained from software advances over this time span, some of it from automated evaluation (and search) tuning, but the greater part from enhanced heuristics for tree search and pruning.[47]

Rajlich himself was certainly a key player in the latter, most notably via the rapid increase in Rybka's strength by about 75 Elo[48] in the 5 months immediately following the initial Dec 2005 release, concluding with Rybka 1.2f in early May 2006.[49] He then added multicore capability in Rybka 2; on a single-core basis, Rybka 2.3.2a in Jun 2007 was about 50 Elo stronger than 1.2f, while Rybka 3 (Aug 2008) jumped 100 Elo above 2.3.2a, mostly (it seems) due to Larry Kaufman's work with the evaluation function. Rajlich has more recently turned to cluster-based initiatives (run on a private rental basis) with Rybka. The current mass-market leader, is Houdini by Robert Houdart, which seems to be greatly inspired by RobboLito[50] in its roots, and with the recent release of Houdini 3, appears to have gained about 100 Elo (or more) since May 2010.

As can be noted, the previous paragraph largely enumerates Elo gains, rather than any specific advances. From comments of Houdart and others, the main focus in engine development today seems to be an accumulation of small gains (say, 1-2 Elo each), validated via large-scale testing. Particularly in conjunction with open source projects, this can make it quite difficult to assess originality. One idea, promoted by Schröder, is to require a minimal Elo gain over a publicly available project and then to accept it as sufficiently different.[51] The exact measurement process and required gain are up to debate, as would what should be done with multiple such forks of the same base. It should also be recalled that the WCCC is a short event with luck playing a large role, and having multiple entries (even if some were 50-100 Elo weaker) would be a distinct advantage.

---

[46] The 1989 paper [CG] already decries the brute force paradigm, strongly stating: *In our opinion the day a 'brute force' machine beats the World Champion will be a great day for hardware engineers, but a loss for AI researchers.*

[47] To what extent (the use of) such heuristics should be termed "artificial intelligence" is a different matter.

[48] The numbers we give follow CCRL lists. However, as with many things in the Fruit/Rybka debate, they can be disputed. For instance, the most venerable of the ratings agencies, the SSDF (`http://ssdf.bosjo.net/list.htm`) puts the difference between Fruit 2.2.1 (Oct 2005) and Rybka 2.3.1 (Jun 2007) at 2831 ± 18 versus 2919 ± 22, or somewhat less than 100 Elo points, on their 32-bit 1200 Mhz Athlons (perhaps a bit outdated by 2007); while the CCRL 40/40 list makes it 2745 ± 13 versus 2919 ± 9, for closer to 175 as the difference.

Also, many opening books in this era were prepared via Rybka analysis; the possible effect of this on the observed results is occasionally discussed, but we know of no experimental data.

[49] Much of this initial Elo gain is already apparent in the mid-March Rybka 1.1 version.

[50] As noted above, RobboLito is based on a reverse-engineering of Rybka 3 (see [24]).

[51] In this regard, *ICCA Journal* 18/1 (page 54) notes that for the 1995 WCCC the programs Virtua and Frenchess had a common origin, but the organising committee felt that the differences were sufficient to consider them as different programs.

10. Conclusion

We have tried to explore the question of move similarity from a variety of angles. In particular, we have taken a large dataset, and analysed it, then extended it, in various ways. Although Fruit/Rybka might have been the impetus for this work when it started, it cannot said to be the only reason why our results could be of interest. However, from the standpoint of originality of engines, we have been asked to catalogue succinctly what our methodology does and does not show, and to note any possibilities of bias that could be induced.

- Our methods are simply to compare move-matching between engines. As indicated in the quotation from Hair in §3.2 above, any statistical measures must take into account that engines do not develop in a vacuum, and that (for obvious reasons) competitors may desire to move-match with a current top engine, via various means.

- The selection of experiments and engines can play a definite aspect in how conclusive any results are. Determining "families" of engines was a difficult consideration already in Hair's original work, and we have tried to take a "reasonable" filtration in §4.2.2, though also keeping the original dataset for comparison. Other experimenters might come to different conclusions.

- Similarly, in the later experiments, one could debate our choice of "top" engines, and also the exclusion basis we used to declare certain engines to be too related.

- The choice of the experiments themselves, that is, the time/depth allotments and whether strength-based scaling was used, is also largely our choice, though some input from others was taken into account. We hope that the concordance metric described in §6.1 gives some indication as to how robust our analysis is.

- In particular regard to the last point, we have indicated alternative methodologies, namely dendrograms in §4.3 and `ponderhit` data in §4.3.1, but we chose not to explore them as much as the Dailey/Hair testing scheme. Although we have noted some presumed current problems and difficulties with these alternative methods, they might yield differing results when reshaped appropriately.[52]

The Fruit/Rybka conclusion from our data is quite inarguable: early Rybka versions, namely from the initial Dec 2005 version and including a version contemporaneous to the June 2007 WCCC, show abnormally large move-matching with Fruit 2.1; this persisted almost no matter what metric was used, and was

[52] We apologise if this is simply a "trivial" statement at the level of philosophy of science, namely that our results are necessarily subject to later rebuttal. It seems that contentious issues always need such reminders, so we explicitly state this.

certainly enough for suspicion to the degree of requesting a source code examination (as per ICGA precedents). Whether or not such move-matching was obtained "permissibly" is obviously beyond the scope of this work. We have, however, tried to fashion some guidelines for tournament directors about how to use the move-matching methodology.

Claims that strong engines have higher move-matching than weaker ones, and similarly when time allotments are increased, are corroborrated by our experiments, though there are still important unanswered questions here, such as what is the "limit" (or asymptote) of move-matching when one increases the time control to a much greater extent (say 3 minutes a move, which is more than 1000 times longer than our experiments).

11. Acknowledgements

The paper has also seen considerable modification and improvement due to input from the anonymous reviewer(s). In some cases, said suggestions were at odds: for instance, one indicated that, due to the contentious nature of Rybka/Fruit, we should not say anything too conclusive – while another wondered what the point was if Rybka/Fruit was not made the centerpiece. We have tried to address most of the issues raised, though as we indicated elsewhere, many of them are beyond what we originally set out to do. In earlier versions of this paper we referred largely to the download package [7] for details (such as which engines were excluded, and exact numbers in many cases), but these are now included in the main text. The amount of data collection has increased more than tenfold since the earliest versions of this paper – again we expect that this will not be enough for some, while others will perhaps admire our assiduousness but inquire whether some of it was really necessary.

Dates for engine releases were often obtained from Internet posts (usually from contemporaneous testers). We have split our references below into published works in the literature, and to those which are less formal in nature.

We refer to subsets of ourselves throughout the paper in the third person, due to the anonymous reviewing process.

References

[B] A. Ban, *Automatic Learning of Evaluation, with Applications to Computer Chess.* Presented at *GAMES 2012*.
`http://www.ratio.huji.ac.il/node/2362`

[CF] P. Ciancarini, G. P. Favini, *Detecting clones in game-playing software.* Entertainment Computing, **1** (2009), 9–15.
`http://dx.doi.org/10.1016/j.entcom.2009.06.001`

[CG] P. Ciancarini, M. Gaspari, *A Knowledge-Based System for the Middle Game.* In *Advances in Computer Chess* **5** (1989), edited by D. F. Beal, published by Elsevier, 219–230.

[DKN] O. David-Tabibi, M. Koppel, N. S. Netanyahu, *Expert-driven genetic algorithms for simulating evaluation functions*. Genetic Programming and Evolvable Machines, **12** (2011), 5–22.
http://dx.doi.org/10.1007/s10710-010-9103-4

[GBT] D. Gomboc, M. Buro, T. A. Marsland, *Tuning evaluation functions by maximizing concordance*. Theor. Comput. Sci. **349** (2005), 202–229.
http://dx.doi.org/10.1016/j.tcs.2005.09.047

[GB] M. Guid, I. Bratko, *Computer Analysis of World Chess Champions.* ICGA Journal, **29** (2006), no. 2, 65–73.

[H] D. Hartmann, *Notions of Evaluation Functions*. In *Advances in Computer Chess* **5** (1989), edited by D. F. Beal, published by Elsevier, 91–141
See also: *How to Extract Relevant Knowledge from Grandmaster Games* (Part 1). ICCA Journal, **10** (1987), no. 1, 14–36.

[HACN] F. Hsu, T. Ananthataman, M. Campbell, A. Nowatzyk, *Deep Thought*. In *Computers, chess, and cognition*, edited by T. A. Marsland and J. Schaeffer, Chapter 5, 55–78, 1990.

[HN] R. M. Hyatt, M. Newborn, *CRAFTY goes deep*. ICCA Journal **20** (1997), no. 2, 79–86.

[LN] D. Levy, M. Newborn, *How Computers Play Chess*. Computer Science Press, 1991.

[RMM] K. W. Regan, B. Macieja, G. McC. Haworth, *Understanding Distributions of Chess Performances*. In *Advances in Computer Games*, 13th International Conference, ACG 2011 (Tilburg), edited by H. J. van den Herik, A. Plaat. Springer LNCS **7168** (2012).
http://dx.doi.org/10.1007/978-3-642-31866-5_20

References

[1] E. Carrisco, *Re: Interview with Vasik Rajlich*. Forum post of Dec. 20, 2005.
http://www.stmintz.com/ccc/index.php?id=471989

[2] Computer Chess Rating Lists (K. Kryukov), *Correlation*.
http://computerchess.org.uk/ccrl/4040/correlation.html

[3] Computer Schaak Vereniging Nederland (C. de Gorter, president), *Similarity test for tournament participation*, 12 March 2012.
http://www.computerschaak.nl/index.php?view=article&id=521

[4] C. Conkie, M. Diosi, F. Quisinsky, A. Schmidt, *Rybka, a new age in Computerchess? Interview with: Vasik Rajlich*, 20 December 2005.
http://www.superchessengine.com/vasik_rajlich.htm

[5] R. Coulom, *2008 Olympiad*. Mailing list post of Aug 2008.
http://tinyurl.com/8d25un4

[6] D. Dailey, *Similarity tester version 03.*
http://komodochess.com/pub/sim03_linux64.zip

[7] D. Dailey, A. Hair, M. Watkins, *Download package for ENTCOM paper.*
http://magma.maths.usyd.edu.au/~watkins/ENTCOM.tar.bz2

[8] FIDE Ethics Comission (R. Rivello, chairman), Decisions issued on July 1, 2012 (Lausanne).
http://tinyurl.com/ca7547s

[9] M. Guid, *WCC Dataset.*
http://www.ailab.si/matej/files/WCC%20Dataset.zip

[10] A. Hair, TalkChess posts, Feb 2012.
http://talkchess.com/forum/viewtopic.php?p=452168

[11] A. Hair, *Elo Increase per doubling*. Forum posts, May 2012.
http://talkchess.com/forum/viewtopic.php?t=43598

[12] N. Hernandez, *Another Conversation with Vasik Rajlich*, July 2011.
http://www.youtube.com/embed/cQshTNJ4pSM

[13] R. Houdart, *Houdini 3 Manual* (see Version History), October 2012.
http://www.cruxis.com/chess/houdini.htm

[14] International Computer Games Association (D. Levy, president), *Disqualification of the program List and its author Fritz Reul*, 27 Nov 2003,
http://www.stmintz.com/ccc/index.php?id=331517

[15] International Computer Games Association (D. Levy, president), *Rybka Disqualified and Banned from World Computer Chess Championships*, 28 June 2011.
http://icga.uvt.nl/?p=56

[16] E. Castillo Jimenez, *Open letter by Eugenio Castillo*. Forum post from F. Quisinsky, Aug 2004.
http://www.stmintz.com/ccc/index.php?id=384790

[17] R. Keene, *Plagiarism in Computer Go*, Scandal of the Month, April 2000.
http://tinyurl.com/9vmjlc3

[18] D. Levy, *Squarknll Chess Program Disqualified*. News item of July 2010.
http://www.grappa.univ-lille3.fr/icga/news_item.php?id=57

[19] R. Murawski, *Clone Engine List*.
http://tinyurl.com/8jmf9ko

[20] R. Pijl, *Open letter to chess programmers*, May 2013.
http://talkchess.com/forum/viewtopic.php?t=47963

[21] S. Schüle, *My recent correspondence with Vasik Rajlich.* Talk Chess forum post, June 2010.
http://talkchess.com/forum/viewtopic.php?p=355707

[22] S. Riis, *A Gross Miscarriage of Justice in Computer Chess*, January 2012.
http://www.chessbase.com/news/2011/riis01.pdf (broken link)
http://tinyurl.com/l89t3vv (web archive)
Concatentation of 4-part series:
http://www.chessbase.com/newsdetail.asp?newsid=7791
http://www.chessbase.com/newsdetail.asp?newsid=7807
http://www.chessbase.com/newsdetail.asp?newsid=7811
http://www.chessbase.com/newsdetail.asp?newsid=7813

[23] M. Uniacke, *HIARCS Chess Strength.*
http://www.hiarcs.com/hiarcs_info.htm

[24] M. Watkins, *A comparison of Rybka and IPPOLIT*, June 2010.
http://open-chess.org/download/file.php?id=13

[25] M. Watkins, Forum post, Dec 2010.
http://www.open-chess.org/viewtopic.php?p=7503#p7503

[26] M. Watkins, *Review of the Rybka case.*
http://magma.maths.usyd.edu.au/~watkins/RECAP.pdf

[27] C. Zhixing, *Evidence of Plagiary of Silver Igo.* Archived webpage from 1999.
http://tinyurl.com/9tu4j7j
See also the Chinese original at http://tinyurl.com/9dl74he

[28] C. Zhixing, *Evidences of the plagiary of Hamlet.* Archived webpage (1999).
http://tinyurl.com/9ggjtuu
See also the Chinese original at http://tinyurl.com/92pjtyw

[29] C. Zhixing, *Plagiary from Handtalk.* Archived webpage from 1999.
http://tinyurl.com/8nzszly
See also the Chinese original at http://tinyurl.com/8eht5tf