

1 Phylourny: Predicting the Knock-out-phase of
2 Tournaments via Phylogenetic Methods by
3 example of the UEFA EURO 2020

4 Ben Bettisworth¹ and Alexandros Stamatakis^{1,2}

5 ¹Computational Molecular Evolution Group, Heidelberg Institute
6 for Theoretical Studies

7 ²Institute for Theoretical Informatics, Karlsruhe Institute of
8 Technology

9 June 24, 2021

Amongst all unimportant
subjects, football is by far the
most important.

Pope John Paul II

10 **Abstract**

11 The prediction of knock-out tournaments represents an area of large
12 public interest and active academic as well as industrial research. Here,
13 we leverage the computational analogies between calculating the so-called
14 phylogenetic likelihood score used in the area of molecular evolution and

15 efficiently calculating, instead of approximating via simulations, the *exact*
16 per-team winning probabilities, given a pairwise win probability matrix P .
17 We implement and make available our method as open-source code and
18 deploy it to calculate the winning probabilities for all teams participating
19 at the knock-out phase of the UEFA EURO 2020 football tournament.
20 We use three different P matrices to conduct predictions, two inferred
21 via our own simple method and one computed by experts in the field.
22 According to this expert P matrix which we trust most, we find that
23 the most probable final is France versus England and that England has
24 a slightly higher probability to win the title. The ability to efficiently
25 and exactly compute winning probabilities, apart from improving and
26 accelerating predictions, might allow for the development of novel methods
27 to compute P .

28 **1 Introduction**

29 Predicting the winner of knock-out (bracket-based/elimination) tournaments
30 can become computationally expensive if a high degree of accuracy shall be at-
31 tained. To fully (and naïvely) evaluate the probability of the final placing of
32 any particular tournament competitor, a polynomial with a comparatively large
33 number of terms must be evaluated (see 3.2 for details). More specifically, for
34 a tournament with n teams, a polynomial with 2^n terms must be evaluated. If
35 one desires to calculate this for every tournament competitor, then n such poly-
36 nomials must be evaluated. Alternatively, one can use stochastic simulations
37 in practice to estimate the probability distribution of the tournament winners.
38 This can potentially be computationally more efficient, but comes at the cost
39 of reduced fidelity of the results [1, 3].

40 However, there exists a similar problem in the field of computational phyloge-
41 netics, that is, the field of Bioinformatics that develops methods for reconstruct-

42 ing the evolutionary histories of currently living species based on their DNA or
43 amino acid sequence data. Computational phylogenetics exhibits a plethora of
44 similarities to the problem of computing the winning probabilities for a tourna-
45 ment. We will henceforth focus on the phylogenetic likelihood model [4] that
46 is currently the most widely used model for phylogenetic inference (i.e., recon-
47 structing evolutionary histories among extant species).

48 Initially, let us consider the problem of computing the likelihood score for
49 a given statistical model of molecular sequence evolution on a given, possible
50 evolutionary history (i.e., a phylogenetic tree). Note that, the specific phyloge-
51 netic tree whose likelihood shall be evaluated also constitutes a parameter of the
52 likelihood model. This tree parameter is special in the sense that it represents
53 the only *discrete* parameter of the phylogenetic likelihood model. While in gen-
54 eral, phylogenetic trees are unrooted, without loss of generality for the purpose
55 of knock-out tournament predictions, we can assume that they are rooted and
56 hence *do* have a direction. In addition, for a tournament, the tree is already
57 given which simplifies the task at hand. The likelihood score on a given tree
58 topology under a given model can be efficiently computed using a dynamic pro-
59 gramming algorithm called 'Felsenstein pruning algorithm' that was presented
60 in Joe Felsenstein's seminal paper that introduces the phylogenetic likelihood
61 model [4].

62 The most striking similarity between the two problems is that computational
63 phylogenetics and knock-out tournament predictions share a directed acyclic
64 graph as a model parameter. Additionally, the shapes of these graphs are re-
65 stricted in analogous ways, which allows to apply computational techniques from
66 phylogenetics to tournament prediction.

67 In addition, both are based on statistical principles. Computational phy-
68 logenetics seeks to compute a likelihood, which is the probability of a model,

69 given some data. So, while this is in principle different than 'just' computing a
70 probability, the underlying structure and order of computations is highly simi-
71 lar. More importantly, the computation of the likelihood can be expressed via
72 polynomials, following a procedure that is essentially analogous to computing
73 the probability of a particular competitor winning a tournament. The above
74 analogies allow us to adapt techniques which have been developed to efficiently
75 compute phylogenetic likelihood scores to also efficiently compute tournament
76 win probabilities.

77 In the following, we propose a novel method of computing win probabilities
78 for a multi-elimination tournament¹, which allows for the exact calculation of
79 win probabilities in conjunction with high computational efficiency. Our novel
80 method, which we call Phylourny, is based on an observation by Ziheng Yang [10]
81 about the aforementioned Felsenstein pruning algorithm. Ziheng Yang points
82 out that Felsenstein's algorithm can be interpreted as an efficient way to com-
83 pute polynomials of a high degree.

84 We implement our new method in a software tool that is also called Phy-
85 lourny. The name is a portmanteau of Phylogeny and tournament. We show
86 that methods which use a similar evaluation strategy as Phylourny are substan-
87 tially faster than naïve tournament prediction approaches. We (will) also assess
88 our method by predicting the winner and winning probabilities of the teams
89 participating at the knock-out phase of the UEFA EURO 2020 European Foot-
90 ball Championship². As already mentioned, a pairwise win prediction matrix
91 P is required as input for our method. We utilize a P matrix based on prior
92 work by experts in the field and two P matrices obtained via a simple method
93 implemented in Phylourny that solely uses match data from the group stage.

¹A multi-elimination tournament is any tournament where a competitor must lose more than once to be eliminated from the tournament. These are almost always double-elimination tournaments, but one can imagine triple or more elimination tournaments. We use this term in order to highlight the more general nature of this method

94 Using these data sources, we obtain three predictions, which are summarized in
95 Section 3.4.

96 2 Background

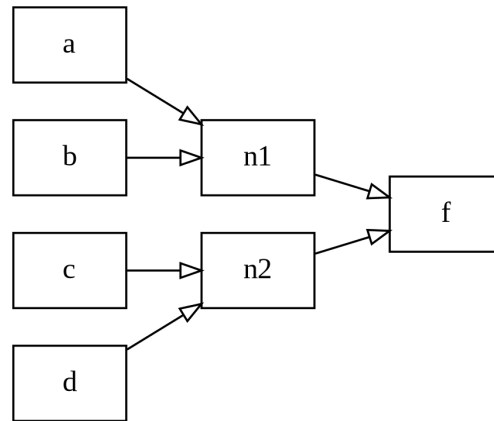


Figure 1: A single elimination tournament with 4 teams.

97 A graph is a set of nodes, and the relationships between those nodes, are
98 called edges. A directed graph is a graph where the edges, here called arcs, have
99 a direction. For example, in Figure 1 there is an arc from **a** to **n1**, but not vice
100 versa. A directed acyclic graph (DAG) constitutes a special, simpler form of
101 a directed graph that does not contain cycles. A graph *has* a cycle if starting
102 from some node, there exists a set of edges which lead back to the same node.
103 Alternatively, one can define a directed graph to be acyclic if, when some node
104 *a* can be reached from *b* this implies that *b* can not be reached from *a*.

105 The number of edges (or arcs) connected to a node is denoted as the degree
106 of the node. The number of arcs pointing *to* a node is called the in-degree, and
107 the number of arcs pointing *away* from a node is called the out-degree.

²The 2020 European Football Championship was postponed to summer 2021 due to the COVID-19 pandemic. This how we can predict a tournament in 2020 with a paper written in 2021.

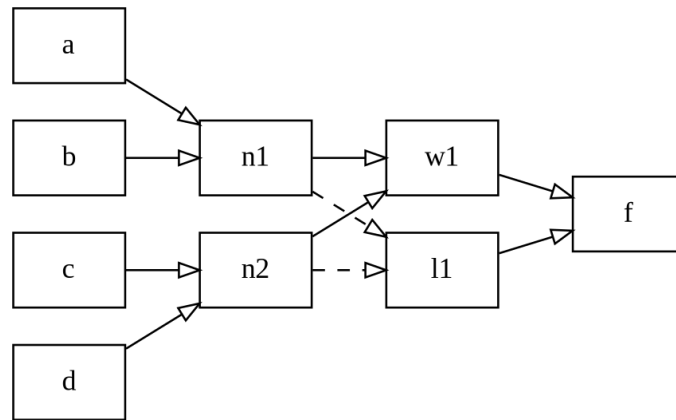


Figure 2: A simple tournament with a losers bracket. The dashed line represents the loser of a specific match. So, in this case, the losers of $n1$ and $n2$ will play each other in the match $l1$.

108 A phylogenetic tree is a DAG with two types of nodes: tips, which have
109 an in-degree of 0; and inner nodes which have an in-degree of 2. Furthermore,
110 almost all nodes in a phylogenetic tree have an out-degree of 1 and only one
111 dedicated node has an out-degree of 0. This particular node is known as the
112 root.

113 A set of events with intrinsic time dependencies, (i.e., one event/match must
114 be completed before another event/match) is naturally acyclic. For instance, in
115 a tournament, the winners of the two semi-finals must be determined before the
116 winner of the final can be determined. Since a tournament is also directed, it is
117 a DAG, as outlined in Figures 1 and 2.

118 In analogy to a phylogenetic tree (for brevity: phylogeny), tournaments have
119 2 types of nodes: competitors which have an in-degree of 0; and matches, which
120 have an in-degree of 2. However, in contrast to a phylogeny, nodes are allowed
121 to have an out-degree of either 1 or 2. The possible out-degree of 2 is to account
122 for the loser of a match moving down to a losers bracket (also called the lower
123 bracket), albeit tournament matches with an out-degree of 1 appear to be more

124 common. Nonetheless, each tournament retains the special node, or match,
125 with an out-degree of 0. This is the final match that will yield the winner of
126 the tournament.

127 The above difference between phylogenies and tournaments introduces a
128 complication. When computing the probability of a specific winner for a par-
129 ticular match, we must account for all possible paths that could have lead the
130 specific winning team to this particular match. Consider the example provided
131 in Figure 2. Here, competitor **a** can either arrive at match **f** via match **w1** or
132 via **l1**. Thus, in order to accurately compute the probability of competitor **a**
133 winning match **f**, we need to sum over the probabilities of arriving at **f** via **w1**
134 or **l1**.

135 Fortunately, if we desire to account for these multiple possible paths, we
136 only need to consider the two matches immediately preceding any given match.
137 However, we need to assume that the probability of winning a match is 'path-
138 independent'. This assumption allows us to 'forget' about the previous matches
139 that a competitor has played and restricts the calculation to the match at hand.
140 Please see Section 3.1 for a comprehensive description of the mathematical de-
141 tails.

142 **3 Method**

143 Initially, we discuss the theory of computing the winner distribution for a sin-
144 gle match in Section 3.1. Subsequently, we discuss how we use this theory
145 to efficiently compute the distribution of winners for a general tournament in
146 Section 3.2. We describe the software tool that implements this method in Sec-
147 tion 3.3. Finally, we outline how we performed our predictions for the UEFA
148 Euro 2020 European Football Championship in Section 3.4.

149 3.1 Theory

150 Initially, we provide some definitions. The win probability vector (WPV) for
151 a given node in the tournament tree is a vector containing the probabilities
152 of observing a given team at that node. We denote the probability of team a
153 winning over team b in a single match as:

$$P_{a+b}$$

154 that is, the probability that 'team a beats team b '. As such, a *general* WPV
155 has n entries, where n is the number of teams in the knock-out tournament.

156 Suppose that we have the most simple tournament with only two teams, a
157 and b . Then, the WPV which describes this tournament is:

$$R_a = P_{a+b}, R_b = P_{b+a} \quad (1)$$

158 Because this constitutes a trivial case, the calculation is straight-forward.
159 To be able to extend this to non-trivial cases, we will artificially complicate the
160 above expression. First, we introduce the WPVs for a and b as w and y . Since
161 a and b are 'tips' of the tree, we can set the probability of observing the team
162 at that node to 1.0 for the team, and 0.0 for all other teams. By doing so, we
163 obtain the expression

$$R_a = (P_{a+b} \times y_a + P_{a+b} \times y_b) \times w_a. \quad (2)$$

164 Further, we define $P_{t-t} := 0.0$ for any team t . So, because $P_{a-a} := 0.0$ and
165 $y_b := 1.0$, we can reduce Equation 2 to Equation 1. Using this property, we
166 can construct a general expression for the WPV at any particular node of a

167 tournament (including the final) with previous matches already computed as

$$R_i = w_i \times \sum_{c \in C} P_{i \rightarrow c} \times y_c, \quad (3)$$

168 where R_i is the i -th entry of the WPV, and C is the set of competitors.

169 For multi-elimination tournaments, we also need to account for the fact that a

170 competitor $c \in C$ can come from both sides of the tournament. Therefore, we

171 need to include a second term in the expression to accommodate the other side:

$$R_i = \left(w_i \times \sum_{c \in C} P_{i \rightarrow c} \times y_{c|i} \right) + \left(y_i \times \sum_{c \in C} P_{i \leftarrow c} \times w_{c|i} \right). \quad (4)$$

172 We calculate $w_{c|i} = w_c / (1 - w_i)$. We interpret this as the probability of

173 observing competitor c at w given that competitor i is the opponent in the

174 match. Thus, Equation 4 is the full general expression for the WPV of a multi-

175 elimination tournament. The final complication is that $P_{a \rightarrow b}$ might be a 'best of

176 k ' series of play-off matches (e.g., in the National Basketball Association (NBA)

177 playoffs). This k can also vary over the duration of the tournament since early

178 matches are often 'best of 1' with $k := 1$, whereas later matches might be 'best

179 of 5' with $k := 5$. We can account for this by introducing a new P' which

180 represents the pairwise probability of winning the 'best of k '.

181 3.2 Implementation

182 In order to compute the most likely winner of the entire tournament, we need

183 to compute the WPV for the final match at the root of the tree. For example,

184 in Figure 1, the match **f** must be evaluated. However, in order to compute

185 this, the corresponding WPVs for matches **n1** and **n2** must be computed, as

186 these represent the intermediate results used in Equation 4. Analogously, for

187 the tournament in Figure 2, the WPVs for matches **n1** and **n2** must be evaluated

188 before the WPV for either match w_1 or l_1 can be computed.

189 Therefore, the tournament matches must be evaluated in the correct tem-
190 poral order to yield a valid result. This sequence of operations on a tree is
191 analogous to how a likelihood score is computed on a phylogeny. As outlined
192 before, the key difference is that a competitor might be able to traverse multiple
193 paths to reach the final match. Instead of finding a simple traversal, we need
194 to find a topological sorting of the tournament DAG. A topological sorting is a
195 list of the nodes of a DAG such that, if the list is read from left to right, all de-
196 pendencies are satisfied. Note that, all DAGs can be sorted topologically [8]. If
197 the DAG is a simple binary tournament tree or phylogenetic tree, a topological
198 sorting can easily be obtained via a post order traversal of the tree. In other
199 words, we can calculate the WPV of the final by computing and storing WPVs
200 bottom up at every node, starting from the leaves/tips of the tree and moving
201 toward its root (the final). This procedure is analogous to the computation of
202 the so-called Conditional Likelihood Vectors (CLVs) on phylogenetic trees via
203 the Felsenstein pruning algorithm.

204 The most important detail missing is how to obtain the pairwise win prob-
205 abilities P . In the preceding Section 3.1, we intentionally considered these
206 probabilities as black boxes for the following two reasons. First, there exist
207 many possible and sophisticated ways to compute P as described, for instance,
208 in recent work by Groll *et al.* [5] or in the classic paper by Dixon and Coles [2].
209 All approaches exhibit advantages as well as disadvantages. Second, computing
210 these probabilities is not the main contribution of this work as we focus on (i)
211 the similarity between phylogenetics and tournaments and (ii) the amount of
212 computations that we can save by applying the Felsenstein pruning algorithm
213 to efficiently and exactly calculate tournament win probabilities, given P .

214 The complexity of a naïve evaluation amounts to $\mathcal{O}(n2^n)$ floating point op-

215 erations. In contrast, the complexity of a Phylourny-like method is $\mathcal{O}(n^2)$. We
216 provide the underlying idea for the time complexity in the following. Consider
217 the probability that competitor 1 wins in an $n := 8$ competitor single elimination
218 tournament. A *single* term for *just one team* is

$$P_{1+2} \times P_{3+4} \times P_{5+6} \times P_{7+8} \times \\ P_{1+3} \times P_{5+7} \times \\ P_{1+5}.$$

219 We have organized the above term into layers, one for each 'tier' of the
220 tournament. Since we start with $n/2$ matches, and halve their number every
221 time, we have a known series of matches which sum to $n - 1$. Now, to count the
222 number of terms, we note that we have a 'choice' for every factor that does not
223 involve competitor 1. For example, we also need to compute the term where
224 competitor 4 beats competitor 3. This means that there are $2^n - \log(n)$ terms.
225 If we combine these, we obtain the total expression $n \times (2^n - \log(n)) = \mathcal{O}(n2^n)$.

226 The naïve space complexity can also be derived from this example. The space
227 required to compute this expression comprises the table of pairwise probabilities,
228 and two additional floating point values. One floating point values is used for
229 the running total of the probability, and the other is used to compute the current
230 term. Therefore, the total space requirement is $n^2 + 2 = \mathcal{O}(n^2)$.

231 The time complexity of a Phylourny-like method is more straight-forward to
232 compute. As we can reuse intermediate results from each match, we solely need
233 to evaluate the WPV for each match once. Additionally, there are n elements
234 in the WPV. Thus, we need to compute n values per $\mathcal{O}(n)$ matches. This yields
235 a time and space complexity of $\mathcal{O}(n^2)$. The space complexity is the same for
236 the naïve method and the Phylourny method as the pairwise win probabilities

237 must be stored.

238 **3.3 Software**

239 A C++ reference implementation of our algorithm is available on GitHub³ under
240 GNU GPL version 3.0. The software only requires CMake to build and also
241 requires git to download. We used this implementation (version v0.1.0) to
242 compute the EURO 2020 predictions presented in Section 3.4.

243 **3.4 Prediction of the UEFA EURO 2020**

244 In order to predict the winner of the UEFA EURO 2020, we need to estimate the
245 pairwise win probabilities of the competing national teams which constitutes a
246 challenging task. However, as we are only interested in verifying our method for
247 *tournaments* we can, for instance, use the match history from the group stages.
248 Therefore, our prediction for the winner of the championship was conducted
249 *after* the group stage, but *before* the knockout stage.

250 Nonetheless, using the match history from the group stage and omitting
251 draws implies that the data are sparse. For instance, for the UEFA EURO
252 2020 tournament, we could only use the results of 10 matches from the group
253 stage. As many teams will not play each other before the elimination stage, the
254 estimation of pairwise win probabilities therefore remains difficult. To overcome
255 this challenge, we deploy two methods. First, we perform a Bayesian sampling
256 of plausible pairwise win probabilities, given the data from the group stage.
257 Second, we utilize the predictive power of existing expert models to infer pairwise
258 win probabilities, which are subsequently used to predict a winner.

³<https://github.com/computations/phylourny>

259 3.4.1 MCMC Sampling

260 The Bayesian sampling is performed via a Markov Chain Monte Carlo (MCMC)
261 search. At each MCMC step, a pairwise win probability matrix is proposed, and
262 the associated WPV is computed for the tournament. Additionally, the likeli-
263 hood of the pairwise win probability matrix (P in earlier sections) is computed
264 using the match data from the group stage. This likelihood represents how likely
265 the proposed pairwise win probability matrix is, given the match history data.
266 This likelihood is different from the phylogenetic likelihood mentioned previ-
267 ously. Informally, a more likely pairwise win probability matrix is one which
268 better explains or fits the previous match history.

269 The MCMC sampling procedure should be continued until the chain has
270 reached 'apparent convergence'. Note that true convergence can only be at-
271 tained if the MCMC sampling is executed infinitely. Further, only the lack
272 of convergence can be assessed via appropriate tools. Hence, as assessing the
273 convergence of MCMC is known to be difficult, we only draw a fixed number
274 of samples. However, computing a single sample using Phylourny is trivial.
275 Therefore, we are able to compute a very large number of samples in a moder-
276 ate amount of time. For a $n := 16$ competitor single elimination tournament,
277 we were able to evaluate 10 million samples in approximately 5 minutes us-
278 ing a high end 2000 EUR laptop. Therefore, predictions for the UEFA 2020
279 knock-out stage were performed using 10 million samples. This corresponds
280 to approximately 33,333 exact calculations of the tournament final WPV per
281 second. We believe that using 10 million samples is justified, as the state space
282 for *this* specific tournament is not excessively large, and should be sufficiently
283 sampled with this number of samples.

284 Our MCMC search is straight forward. Each P_{a+b} is proposed according
285 to a uniform prior, with $P_{b+a} = 1 - P_{a+b}$ for all competitors a and b . We

286 sample every proposal, and we run the search until we have obtained 10 millions
287 samples. When computing the summary statistics, we discard the first one
288 million samples as burn in.

289 Once we have obtained all samples from the MCMC procedure, we can
290 compute two predictions: the maximum likelihood prediction (MLP), or the
291 maximum marginal posterior prediction (MMPP). The MLP is simply the pre-
292 diction given by the pairwise win probability matrix with the highest likelihood
293 score, whereas the MMPP is the average prediction from all samples. Because an
294 MCMC search will sample the posterior with a probability distribution hopefully
295 approximating the true posterior, the average over all samples is approximately
296 the average of the posterior. The difference between these two predictions is one
297 of philosophical nature rather than mathematics, as they encapsulate distinct
298 interpretations about what 'really' matters. The school of thought advocating
299 the MLP claims that the only thing that matters is the *most likely* outcome,
300 regardless of the underlying distribution, whereas the school of thought sup-
301 porting the MMPP claims that the *totality of evidence* is what matters. A
302 discussion about the merits of these two schools of thought is beyond the scope
303 of this paper.

304 **3.4.2 Model Based Forecast**

305 To perform a model based prediction, we use an existing model (we call this
306 the Lazy Method (LM)), published by Groll *et al.* [5], who have also published
307 previous football tournament predictions, for instance, for the Woman's World
308 cup in 2019 [6]. Groll *et al.* deploy a random forest approach, utilizing match
309 histories, bookmaker odds, and average player ratings to obtain a pairwise win
310 probability matrix as well as general predictions for the UEFA EURO 2020
311 tournament. We have thus used their pairwise win probability matrix which

312 was published on the web⁴, and used Phylourny to compute the WPV of the
313 tournament.

314 The pairwise win probabilities from LM are input directly to Phylourny and
315 used to compute the WPV. We performed no modifications to the data, other
316 than to remove the teams that do not participate at the knock-out stage. This
317 P matrix is the one we trust most due to the broad input data from distinct
318 sources being used and the tournament prediction track record of the associated
319 research group.

320 4 Results

321 Overall, we computed 3 predictions for the three alternative pairwise win proba-
322 bility calculations: Lazy Method (LM), Maximum Likelihood Prediction (MLP),
323 and Maximum Marginal Posterior Prediction (MMPP). Phylourny was executed
324 as follows to calculate the predictions:

```
325 ./phylourny --teams euro2020.ini --matches euro-match-history.csv  
326 --probs euro.probs.csv --prefix EUR02020
```

327 The respective input data and relevant output files of Phylourny are available
328 at [https://cme.h-its.org/exelixis/resource/download/phylourney-data.](https://cme.h-its.org/exelixis/resource/download/phylourney-data.tar.bz2)
329 `tar.bz2`.

330 The predicted tournament winner and loser of the final for each of the three
331 methods is summarized in Table 1.

332 Beyond the discrete predicted winner, the scientifically more interesting re-
333 sult is the full WPV for LM, MLP, and MMPP as predicting football matches
334 is know to be notoriously difficult because of the low number of goals being
335 scored that induces a substantial impact of chance onto the final result [7]. To

⁴<https://www.zeileis.org/news/euro2020/>

Prediction Method	Winning Team	Losing Team
LM	England	France
MLP	Netherlands	France
MMPP	Netherlands	Italy
Actual Result	TO BE INSERTED on July 11	

Table 1: Table of pairwise win prediction methods, the team winning the tournament and the team losing the final induced by the respective method, and the actual result.

336 this end, football match predictions do exhibit a high degree of uncertainty and
337 winner predictions should thus be displayed as per-team probabilities, that is,
338 as WPVs. We include the respective WPVs under LM, MLP, and MMPP in
339 Table 4. The two teams with the highest probability of winning the tournament
340 per pairwise prediction method/matrix P are shown in bold font. These data
341 can be used to assess the prediction accuracy of our method in retrospect, once
342 the tournament is over on July 11, 2021.

343 Comparing our prediction using the LM method with that of the original
344 paper describing the LM method [5] which we denote as oLM we obtain: 16.5%
345 for England (oLM: 13.5%), 14.9% for France (oLM: 14.8%), 10.2% for Spain
346 (oLM: 12.3%), 10.0% for Portugal (oLM: 10.1%), and 11.4% for Germany (oLM:
347 10.1%). The slight deviations in the predictions despite using the exact same P
348 matrix are due to the fact that the oLM values were computed *before* the group
349 phase including the prediction of the by then still *unknown* tournament tree for
350 the elimination phase. In contrast to this, our predictions were computed *after*
351 the group phase for a *known* tournament tree.

352 5 Conclusion

353 We have shown that the problem of predicting tournament winners is suffi-
354 ciently similar to phylogenetic likelihood calculations such that analogous com-
355 putational techniques can be applied. We have demonstrated this by developing

Pred. Meth.	Bel	Por	Ita	Aus	Fra	Swi	Cro	Spa	Swe	Ukr	Eng	Ger	Net	Cze	Wal	Den
LM	0.057	0.100	0.076	0.011	0.149	0.016	0.025	0.102	0.018	0.006	0.165	0.114	0.077	0.007	0.011	0.064
MLP	0.100	0.004	0.159	0.000	0.192	0.002	0.001	0.094	0.089	0.000	0.103	0.025	0.225	0.000	0.001	0.004
MMPP	0.074	0.047	0.080	0.058	0.077	0.050	0.052	0.064	0.066	0.041	0.083	0.061	0.088	0.050	0.054	0.053

Table 2: Complete WPVs for the knock-out phase of the UEFA Euro 2020 comprising all 16 teams using the LM, MLP, and MMPP methods for predicting pairwise win probabilities. The two teams with the highest probability of winning the tournament per method are shown in bold font.

356 methods inspired by computational phylogenetics to predict tournaments, and
357 that applying these methods yields substantial computational speedups in terms
358 of theoretical run time complexity. In addition, we can calculate the final WPV
359 of a tournament exactly instead of using simulations to approximate it. This
360 also allows, for instance, for a seamless deployment of MCMC methods such as
361 illustrated by our admittedly very simple example in Section 3.4.1.

362 Furthermore, we demonstrate the practicality of these new methods by im-
363 plementing them into a new software tool called Phylourny.

364 As we are writing this *before* the tournament enters its knock-out phase,
365 we do not know how successful our method will be at predicting the true out-
366 come⁵. Nonetheless, we can already discuss the two theoretical shortcomings of
367 our approach regardless of the success of our prediction. First, the prediction
368 'difficulty' is predominantly deferred into estimating the pairwise win proba-
369 bility matrix. This constitutes the central problem of tournament prediction,
370 which we do intentionally not directly address. Betting companies with their
371 substantial resources and other researchers have already addressed this prob-
372 lem to a large extent [9]. Instead, we present a *computational* method, which
373 will accelerate the exact computation of final win probabilities, given some es-
374 timation of pairwise win probabilities, and a surprising connection between two
375 seemingly unrelated branches of science.

376 Second, the assumption of path independence might not be true, as com-
377 petitors might suffer from fatigue from competing in more matches, if a team
378 must proceed through the lower bracket in order to reach the finals or by having
379 to play harder opponents or play over-time. Furthermore, other 'intangibles',
380 such as moral or confidence, are hard to quantify, also questioning the path
381 independence assumption. Nonetheless, this path dependence can be addressed
382 via a more involved method of calculating the pairwise win rate matrix, as one

383 can also deploy a match-dependant P matrix.

384 Despite the two deficiencies mentioned above, we have shown that we can
385 compute, both exactly and efficiently, the WPV for a tournament. This is im-
386 portant because, many advanced methods of analysis require exact results to
387 be applicable. For example, when sampling from a posterior using an MCMC
388 search, it is desirable to have an accurate result for each sample. While a suffi-
389 cient degree of accuracy can be obtained via an appropriately large number of
390 simulations, this approach is computationally expensive and might even become
391 prohibitive. We demonstrate that we can efficiently conduct such an analysis
392 by implementing our own (naïve) MCMC analysis of the UEFA EURO 2020
393 football tournament.

394 While we consider this work as being complete, there exist further areas of
395 investigation that can be explored. An example is exploring the 'stability' of
396 complicated tournaments by slightly perturbing P and examining the result-
397 ing probabilistic outcome. Due to the increased computational efficiency and
398 the ability of Phylourny to *exactly* calculate the final WPV, such studies are
399 substantially more tractable now. Another area of interest would be to further
400 develop the MCMC sampling. We currently use an extremely naïve MCMC
401 search that could become more efficient by specifying more elaborate methods
402 for proposing new parameters.

403 **References**

- 404 [1] DEMSYN-JONES, R. Misadventures in Monte Carlo. *Journal of Sports*
405 *Analytics* 5, 1 (Mar. 2019), 1–15.
- 406 [2] DIXON, M. J., AND COLES, S. G. Modelling association football scores and
407 inefficiencies in the football betting market. *Journal of the Royal Statistical*

- 408 *Society: Series C (Applied Statistics)* 46, 2 (1997), 265–280.
- 409 [3] EKSTRØM, C. T., VAN EETVELDE, H., LEY, C., AND BREFELD, U.
410 Evaluating one-shot tournament predictions. *Journal of Sports Analytics*
411 7, 1 (Apr. 2021), 37–46.
- 412 [4] FELSENSTEIN, J. Evolutionary trees from dna sequences: a maximum
413 likelihood approach. *Journal of molecular evolution* 17, 6 (1981), 368–376.
- 414 [5] GROLL, A., HVATTUM, L. M., LEY, C., POPP, F., SCHAUBERGER, G.,
415 VAN EETVELDE, H., AND ZEILEIS, A. Hybrid Machine Learning Forecasts
416 for the UEFA EURO 2020. *arXiv:2106.05799 [cs, stat]* (June 2021). arXiv:
417 2106.05799.
- 418 [6] GROLL, A., LEY, C., SCHAUBERGER, G., EETVELDE, H. V., AND
419 ZEILEIS, A. Hybrid machine learning forecasts for the fifa women’s world
420 cup 2019, 2019.
- 421 [7] HEUER, A., MÜLLER, C., AND RUBNER, O. Soccer: Is scoring goals a
422 predictable Poissonian process? *EPL (Europhysics Letters)* 89, 3 (Feb.
423 2010), 38007. Publisher: IOP Publishing.
- 424 [8] KAHN, A. B. Topological sorting of large networks. *Communications of*
425 *the ACM* 5, 11 (1962), 558–562.
- 426 [9] KAUNITZ, L., ZHONG, S., AND KREINER, J. Beating the bookies with
427 their own numbers - and how the online sports betting market is rigged.
428 *arXiv:1710.02824 [cs, stat]* (Nov. 2017). arXiv: 1710.02824.
- 429 [10] YANG, Z., ET AL. *Computational molecular evolution*, vol. 284. Oxford
430 University Press Oxford, 2006.