Bioinformatics Advance Access published May 16, 2011

SCORER 2.0: An algorithm for distinguishing parallel dimeric and trimeric coiled-coil sequences

Craig T. Armstrong [1 †] and Thomas. L. Vincent [1,2 †], Peter J. Green [3 *] and Derek N. Woolfson [1,4 *]

[1]School of Chemistry, University of Bristol, Bristol, BS8 1TS.
[2]Bristol Centre for Complexity Science, University of Bristol, Bristol, BS8 1TR.
[3]Department of Mathematics, University of Bristol, Bristol, BS8 1TW.
[4]School of Biochemistry, Medical Sciences Building, University of Bristol, Bristol, BS8 1TD.

Associate Editor: Prof. Alfonso Valencia

ABSTRACT

Motivation: The coiled coil is a ubiquitous α-helical protein-structure domain that directs and facilitates protein-protein interactions in a wide variety of biological processes. At the protein-sequence level, coiled coils are quite straightforward and readily recognised via the conspicuous heptad repeats of hydrophobic and polar residues. However, structurally they are more complicated, existing in a range of oligomer states and topologies. Here we address the issue of predicting coiled-coil oligomeric state from protein sequence.

Results: The predominant coiled-coil oligomer states in Nature are parallel dimers and trimers. Here we improve and retrain the first-published algorithm, SCORER, that distinguishes these states, and test it against the current standard, MultiCoil. The SCORER algorithm has been revised in two key respects: First, the statistical basis for SCORER is improved markedly. Second, the training set for SCORER has been expanded and updated to include only structurally validated coiled coils. The result is a much-improved oligomer-state predictor that outperforms MultiCoil, particularly in assigning oligomer state to short coiled coils, and those that are diverse from the training set.

Availability: SCORER 2.0 is available via a web-interface at http://coiledcoils.chm.bris.ac.uk/Scorer. Source code, training sets and Supporting Information can be downloaded from the same site.

Contact: D.N.Woolfson@bristol.ac.uk or P.J.Green@bristol.ac.uk

1 INTRODUCTION

The coiled coil is a protein-structure domain comprising two or more α-helices wound around each other, usually in a left-handed fashion (Crick, 1953; Lupas and Gruber, 2005)(Fig. 1). By using the SUPERFAM method to detect coiled-coil containing superfamilies of proteins, it has been estimated that on average 2.9% of open reading frames across all genomes contain regions that encode coiled coils (range, 0.3– 6.5%) (Rackham *et al.*, 2010). Moreover, coiled-coil domains play roles in mediating protein-protein interactions across a wide array of biological functions from transcription, through membrane remodeling, to cell and tissue structure and stability. Despite this ubiquity and diversity, a relatively straightforward sequence motif of hydrophobic (H) and polar (P) residues HPPHPPP underlies most coiled-coil structures. The positions in these so-called heptad repeats are labeled *a* through *g*, with the hydrophobic sites falling at the *a* and *d* positions. Traditionally, it is these repeats that are identified by coiled-coil-region prediction algorithms (*vide infra*). However, this apparent simplicity of coiled-coil sequences hides considerable complexity in their 3D structures: coiled-coil assemblies can have different numbers of helices, which may be in parallel or anti-parallel arrangements, and may be formed from the same (homo) or different (hetero) helical sequences (Lupas and Gruber, 2005; Moutevelis and Woolfson, 2009). Coiled-coil-structure prediction, then, can be aimed at one or more of three problems:

1. Given a protein sequence, can we accurately identify coiled-coil regions?
2. Given a coiled-coil sequence, can we correctly assign its architecture and topology?
3. Given two or more coiled-coil sequences, can we predict how these combine to form functional assemblies?

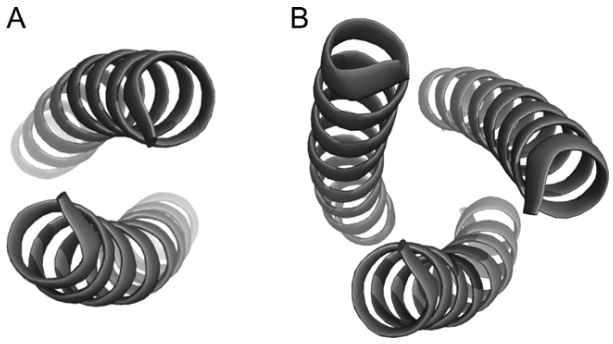

Fig. 1: Cartoon representations of (A) a dimeric and (B) a trimeric coiled coil shown from the ends of the helices (PDB identifiers 1KD9 and 1BB1, respectively).

*To whom correspondence should be addressed.
†Authors contributed equally to this paper.

Here we focus on the second problem, and specifically on oligomer-state prediction. However, the first problem of locating coiled-coil regions in protein sequences *per se* is also pertinent here. This is because the identification of coiled-coil regions is a prerequisite to predicting coiled-coil oligomer state.

Several algorithms exist to tackle the first problem: the widely used COILS (Lupas *et al.*, 1991) utilises residue frequencies at different positions of the heptad repeats (Parry, 1982) of known coiled-coil structures to predict whether new sequences are coiled coils or not. PAIRCOIL (Berger *et al.*, 1995) — and its successor, PAIRCOIL2 (McDonnell *et al.*, 2006) — builds upon this method by utilising correlations in amino acid usage at the different heptad positions. Algorithms such as MARCOIL (Delorenzi and Speed, 2002) and CCHMM (Fariselli *et al.*, 2007) find coiled-coil regions using Hidden Markov Models (HMMs). Some more recent algorithms attempt to incorporate evolutionary information into coiled-coil search strategies: CCHMM-PROF (Bartoli *et al.*, 2009) and PCOILS (Gruber *et al.*, 2006) — the sequels to CCHMM and COILS, respectively — subject query sequences to rounds of PSIBLAST searches, and predict coiled-coil regions from profiles made from these searches. SOSUIcoil (Tanizawa *et al.*, 2008) uses physico-chemical parameters — such as predicted sequence amphiphilicity — in conjunction with amino-acid propensities to predict coiled-coil regions, including breaks or non-canonical patches within them. SPIRICOIL (Rackham *et al.*, 2010) incorporates coiled-coil-containing proteins into the SUPERFAM database (Gough *et al.*, 2001), and predicts coiled-coil regions by comparison with homologous proteins of known structure. Spiricoil has been shown to perform better than the other algorithms when predicting coiled-coil regions, but its reliance on structurally resolved homologues is limiting. Of the truely *ab initio* coiled-coil prediction algorithms, MARCOIL and PCOILS are thought to perform best (Gruber *et al.*, 2006), although SOSUIcoil, CCHMM, and CCHMM-PROF are yet to be independently benchmarked.

Three algorithms exist to tackle the architecture problem: SCORER (Woolfson and Alber, 1995), MultiCoil (Wolf *et al.*, 1997), and the aforementioned SPIRICOIL (Rackham *et al.*, 2010). SCORER uses a log-odds-based scoring system to distinguish whether coiled-coil sequences are more similar to a profile derived from parallel dimeric coiled coils, or a profile derived from parallel trimeric coiled coils. The MultiCoil algorithm is a hybrid of 2 algorithms: PairCoil is used to predict coiled-coil regions, and differences in pairwise residue correlations in known parallel dimeric and parallel trimeric coiled-coils are then used to assign oligomeric state. Again, SPIRICOIL assigns oligomeric state based on homology to proteins of known 3D structure. SPIRICOIL has been shown to outperform other methods, and has the advantage of being able to predict higher-order coiled-coil architectures, *i.e.*, it is not limited to dimers and trimers. However its use is voided when dealing with proteins with no structurally resolved homologues. Of the *ab initio* methods, MultiCoil has enjoyed the most popularity.

A large body of work has been performed to address the issue of how partner selection is determined in coiled coils. In particular, the interactions that stabilise dimeric assemblies of coiled coils have been analysed using experimental and bioinformatic methods (Krylov *et al.*, 1998; Newman and Keating, 2003; Acharya *et al.*, 2006; Mason *et al.*, 2006; Hadley *et al.*, 2008; Steinkruger *et al.*, 2010; Reinke *et al.*, 2010), and rules gleaned from these analyses have been used to design sets of mutually exclusive coiled-coil dimers (Bromley *et al.*, 2009; Reinke *et al.*, 2010).

Given the abundance of observed and possible coiled-coil architectures and topologies (Walshaw and Woolfson, 2001; Lupas and Gruber, 2005; Moutevelis and Woolfson, 2009) and the current limits of homology based coiled-coil prediction such as SPIRICOIL, one of our focuses has been on improving *ab initio* methods for coiled-coil oligomer-state prediction. Both of the aforementioned algorithms, MultiCoil and SCORER were written in the 1990s, and neither has been updated since. Although both SCORER and MultiCoil are limited to the prediction of parallel dimers and trimers, these structures represent $\sim 50\%$ of known coiled-coil structures (Moutevelis and Woolfson, 2009). Thus, attempts to update these algorithms seemed like a logical step towards the goal of better and broader (i.e., multi-state) coiled-coil predictors.

Here we present SCORER 2.0, a significantly revised and updated version of the SCORER algorithm, which uses advanced statistical methods and is trained on a pristine set of coiled-coil sequences of known 3D structure. The latter were culled from the RCSB PDB (Berman *et al.*, 2000) using SOCKET (Walshaw and Woolfson, 2001). The SOCKET algorithm finds the knobs-into-holes packing between coiled-coil helices that is dictated by the underlying heptad sequences repeat. Application of SOCKET to the RCSB PDB has rendered CC+ (Testa *et al.*, 2009), a database of all known structurally resolved coiled coils. It was from CC+ that the pristine set was ultimately selected. SCORER 2.0 classifies coiled-coil sequences of unknown oligomeric state by using statistically significant differences in the frequencies of the 20 proteogenic amino acids at the 7 heptad positions in dimer and trimer profiles. This is achieved by using a Bayes factor method, which accounts for the uncertainty that may arise in profile tables. Finally, SCORER 2.0 was compared to MultiCoil using a variety of PAIRCOIL parameters, circumventing the issue of MultiCoil having an obligatory PAIRCOIL front-end. SCORER 2.0 is available online, and has the option of being used in conjunction with a MARCOIL front-end.

2 METHODS

2.1 Coiled-coil training and test sets

The sequences of parallel dimeric and parallel trimeric canonical — that is, heptad based — coiled coils longer than 14 residues in length were obtained from the CC+ database (Testa *et al.*, 2009), aligned using Clustalw2 (Larkin *et al.*, 2007) (maximum gap penalties were used to conserve the alignment of the heptad repeat), and then culled using CD-HIT (Li and Godzik, 2006) at redundancy cutoff intervals of 5% in the range 40% – 95%. The corresponding structures were validated to ensure no coiled coils were part of higher-order assemblies. We named the resulting set of structures corresponding to 50% maximum identity cutoff the *pristine dataset*. The identity threshold of $25 - 30\%$, often used for culling protein datasets, is too restrictive for coiled-coil sequences, which have a restricted amino acid usage, and therefore regarded as regions of low complexity (this is addressed in greater depth in the Supporting Information to this manuscript). The pristine dataset comprised 133 dimeric and 33 trimeric coiled-coil sequences. Position-specific Scoring Matrices, *PSSM*s (Parry, 1982), were derived for both dimer and trimer sequences by counting the occurrence of the 20 proteogenic amino acids at each of the heptad positions, yielding two 20×7 tables. Each element of these tables was denoted $PSSM_{o,a,r}$ where o can take the values 2 or 3 to denote the dimer and trimer *PSSM*s,

respectively; a represents each of the 20 proteogenic amino acids in standard single-letter code; and r denotes the heptad register. The total number of counts across each register is denoted $TOT_{o,r}$.

A dataset of divergent dimeric and trimeric coiled-coil sequences was also created. From the full list of dimers and trimers available to us, any sequence with below 40% identity to any other entry was labeled as divergent. Pairwise identity between each sequence was computed using the Smith-Waterman algorithm (Smith and Waterman, 1981) implemented in the EMBOSS suite (Rice *et al.*, 2000). The final *divergent dataset* comprised 95 dimeric and 25 trimeric coiled-coil sequences.

Full details are given at http://coiledcoils.chm.bris.ac.uk/Scorer.

2.2 Scoring

The original SCORER algorithm reports the relative likelihood that a test coiled-coil sequence is representative of a dimer or a trimer profile using a log-likelihood ratio:

$$Score = \sum_{i=1}^{l} \log \frac{PROF_{2,a_i,r_i}}{PROF_{3,a_i,r_i}} \tag{1}$$

where $A = \{a_1, ..., a_l\}$ and $R = \{r_1, ..., r_l\}$ represents the amino-acid residues and associated register positions of the test coiled coil sequence S, with residues numbered $i = 1, 2, \ldots, l$ and observed oligomeric state $O = \{2, 3\}$.

By selecting the terms in the dimer and trimer *PSSM*s for which the values were significantly different, and using only those to discriminate between dimeric and trimeric coiled coil sequences, the SCORER algorithm achieves a good rate of prediction. However, this method of scoring contains a few unsatisfactory features that we propose to resolve in SCORER 2.0:

1. The original SCORER algorithm uses a decision threshold of 0 to classify coiled coils, and does not take into account the prior odds of dimer and trimer occurrence in the user's study. While this would not be a problem if the odds of dimers vs trimers were 1:1, we know that this is typically not the case, with dimers being far more common than trimers (133:33 in our database for example); the background probabilities of dimers and trimers should be accounted for. Accordingly, the log prior odds of dimer vs trimer relative to the users experiment should be added to the score, and the subsequent result can then be properly interpreted as the log posterior odds, and used to make a decision.
2. The original SCORER method makes no allowance for errors in estimation in the profile probabilities $PROF_{o,a,r}$ for example, if a particular residue a is rare for a particular oligomeric type o and register r then $PROF_{o,a,r}$ will have a large associated standard error and may even be estimated incorrectly to be zero, potentially skewing the results. The statistical analysis in the original SCORER algorithm circumvented this problem by only including *PSSM* values for those amino acids that made up at least 5% of the residues at a given site in both the dimer and trimer databases. As a consequence, amino acids that were poorly represented at a particular heptad position in one dataset, but reasonably well represented in the other were ignored, and some amino acids that could contribute to oligomer discrimination may have been overlooked. Rather than suppressing insignificant or poorly populated terms, it is better to modify the score in a principled way that has the effect of diluting the influence of poorly estimated profile values.

Assuming that the probabilities of obtaining amino acid a for any combination of oligomeric state o and register r are independent, SCORER 2.0 assigns a score to an amino-acid sequence according to:

$$Score_{2.0} = \log \prod_{r=1}^{7} \left[\frac{(TOT_{3,r} + 20\delta)^{(y_r)}}{(TOT_{2,r} + 20\delta)^{(y_r)}} \times \prod_{a=1}^{20} \frac{(PSSM_{2,a,r} + \delta)^{(x_{a,r})}}{(PSSM_{3,a,r} + \delta)^{(x_{a,r})}} \right] \tag{2}$$

where $x_{a,r}$ is the number of a residues at register r in the test sequence, $y_r = \sum_a x_{a,r}$ is the total number of amino acids at register r in the test sequence, and $m^{(x)}$ stands for the rising factorial symbol $m(m+1)(m+2)\ldots(m+x-1)$. The constant δ is a prior parameter that provides stability in estimation. By introducing this parameter, we adjusted the probability of rare, but not impossible events, artificially so that no probability is estimated as zero; as can be seen, the relative impact of adding δ is negligible on cells with large counts. Cross-validation indicated that a value of $\delta = 1$ provided optimal performance; this corresponds to an uninformative (uniform) prior assumption.

2.3 MultiCoil

MultiCoil uses PAIRCOIL to locate coiled-coil regions in protein sequences, and then assigns whether each residue deemed to be in a coiled-coil conformation is part of a dimeric or trimeric assembly. As a consequence, it is not possible to uncouple the coiled-coil region and oligomeric-state predictions, and known coiled-coil sequences can only be assigned an oligomeric state if they are recognised by PAIRCOIL. Also, coiled-coil regions submitted to MultiCoil as part of a full native protein chain may be truncated or extended depending on where PAIRCOIL assigns the domain boundaries. Nonetheless, with no alternative, coiled-coil containing protein sequences were submitted to the publically available MultiCoil web server, using a PERL script. For each amino-acid residue, a, in a protein sequence, S, MultiCoil assigns a coiled-coil probability, C_a, and oligomeric-state scores, D_a and T_a (dimer and trimer scores, respectively), where $C_a = D_a + T_a$. For the purpose of this work, this method of scoring was converted into a single oligomeric-state score:

$$S_a = \frac{D_a - T_a}{C_a} \tag{3}$$

Thus, positive score will indicate a dimeric prediction, while a negative score will indicate a trimeric prediction. This conversion of the MultiCoil scores is necessary, as it allows the performance measures discussed in the next section. The conversion does not impact the performance of MultiCoil in any way, and simply represents an alternative method of displaying the MultiCoil output.

2.4 Assessing the performance

The performance of both SCORER 2.0 and MultiCoil were compared using Receiver Operator Characteristic (ROC) curves (for example, see (Fawcett, 2006)). ROC curves are plots of the True Positive Rate (TPR) as a function of the False Positive Rate (FPR). The True Positive Rate is the probability of correctly classifying a true instance and is defined as:

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

The False Positive Rate is the probability of assigning a false instance as true, and is defined as:

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

Here, TP, FN, FP and TN represent counts of true positives, false negatives, false positives and true negatives, respectively. Data points are plotted in descending order of confidence; that is, the most confident predictions are plotted first and occur nearer to the origin. One advantage of this method is that the Area Under the Curve (AUC) lies between zero and unity, and gives a metric of how well a prediction algorithm separates the data sets. In the hypothetical case of the perfect separation of two data sets, 100% of true positives would be identified without the occurrence of a single false positive, for which a ROC curve depicting this would yield a single point at [0,1] and an AUC of 1. All AUC values and ROC curves were generated using the ROCR package freely available in the R software (Sing *et al.*, 2005).

3 RESULTS

3.1 Optimal redundancy cutoff in the SCORER 2.0 training set

As far as possible, sequence similarity or redundancy between training and test sets should be eliminated in the assessment of prediction software to prevent returning artificially high accuracies. A difficulty arises here with coiled-coil sequences, however, as they share the heptad repeat, which increases the potential for similarity even in the absence of homology. Therefore, we assessed how SCORER 2.0 predicted a number of divergent sequences after being trained on training sets culled at different redundancy cutoffs. This investigated whether there was an optimal redundancy cutoff for the training set, and assessed how robust the SCORER 2.0 algorithm was to the inclusion or deletion of training data. This is analogous to the bias-variance tradeoff (Geman *et al.*, 1992), which looks at how the introduction of a certain amount of bias in an otherwise unbiased estimator may improve its performance. This is of relevance here as we anticipate retraining SCORER 2.0 as more sequence and structural data become available. For each sequence redundancy cutoff in a training set, an AUC score was obtained for divergent sequences of different lengths. Regardless of redundancy in the training set, AUC scores for sequences longer than 14, 21 and 28 residues in the test set were found to be in the range of 0.8 ± 0.01, 0.89 ± 0.02, 0.94 ± 0.003 respectively. The results showed SCORER 2.0 to be robust to changes in its training set, and that it provided comparable predictions and performance for any redundancy cutoff above 40% (Fig. S1, Supporting Information).

3.2 Comparison with the original SCORER algorithm using a pristine set of coiled-coil sequences

The original SCORER algorithm assigned scores to $103/133$ dimers and $30/33$ trimers of the pristine set of coiled-coil sequences used in this new work; the remaining sequences were not scored as they did not contain features deemed to discriminate between dimer and trimer formation according to the significance cut-off criteria defined in the original SCORER paper (Woolfson and Alber, 1995). For all the dimeric and trimeric sequences that could be assigned a score, SCORER achieved an AUC value of 0.63. SCORER 2.0 scored all of the sequences, yielding an AUC of 0.77. When restricted to analysing only the sequences that the original SCORER algorithm assigned, SCORER 2.0 achieved an AUC of 0.76 (Fig. S2, Supporting Information). Thus in all cases SCORER 2.0 offers a distinct improvement in performance over SCORER.

3.3 Comparison with MultiCoil on a pristine set of coiled-coil sequences

The abilities of SCORER 2.0, SCORER and MultiCoil to predict accurately the oligomeric state of the coiled coils in the pristine dataset (see Methods) were compared. SCORER 2.0 and SCORER were assessed using leave-one-out cross-validation to provide independent tests of the utility of the algorithm. These data differ from those discussed in section 3.2 due to the fact that MultiCoil can only score sequences longer than 21 residues. The results of these tests are shown as ROC curves in Figure 2.

The AUC values and ROC curves in Figure 2 show that SCORER 2.0 achieves a better discrimination rate of coiled-coil oligomeric state than SCORER and MultiCoil (0.86 vs 0.75 and 0.63, respectively). We found the improvement over MultiCoil to be particularly marked for short dimeric coiled coils, which is important as this oligomeric state accounts for a large proportion of the total coiled-coil population. However, the results reported in this section reflect the performance of SCORER 2.0 and MultiCoil under somewhat contrived conditions: all the data were obtained from experimental and SOCKET-derived annotations. In real-life predictions, most coiled-coil sequence data will not be as well delimited and defined as our training set, since coiled-coil-region prediction software rather than high-resolution structures will provide the input data. To ensure a fair comparison between SCORER 2.0 and MultiCoil, the PAIRCOIL predicted regions, along with their register assignments, were used by SCORER 2.0 in place of the known SOCKET-derived coiled-coil regions. For these reasons, we suggest that the results presented in the next section are a more representative comparison test of SCORER 2.0 and MultiCoil.

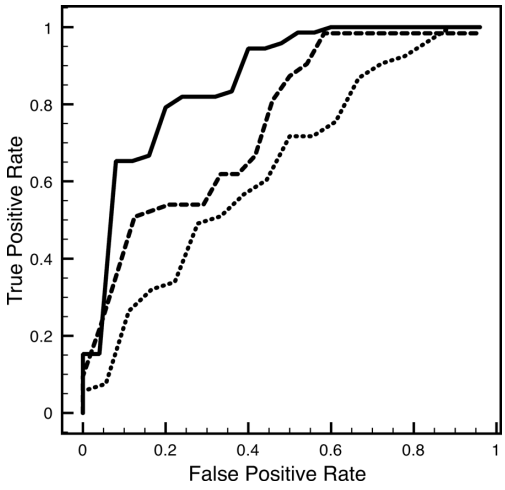

Fig. 2: ROC curves of SCORER 2.0, SCORER and MultiCoil when used to classify the oligomeric state of coiled coils using leave-one-out cross-validation in our pristine test set. Only coiled coils with sequence > 20 amino acids were used, as MultiCoil will not accept any input shorter than 21 characters. Solid line, SCORER 2.0; dashed line, SCORER; dotted line, MultiCoil. AUC for SCORER 2.0: 0.86; AUC for SCORER: 0.75; AUC for MultiCoil: 0.63. Test set comprised 72 dimeric sequences and 25 trimeric sequences.

3.4 SCORER 2.0 vs MultiCoil, using a PAIRCOIL front-end

As mentioned above, comparing SCORER 2.0 and MultiCoil is complicated, as MultiCoil's oligomeric-state and coiled-coil region prediction function are coupled. Conversely, SCORER 2.0 requires input of a sequence thought to be a coiled coil, and its corresponding register, and will return an oligomer-state prediction regardless of whether there is a true coiled-coil present or not. Put another way, the performance of the two algorithms depends on the front-end coiled-coil-region predictions, and these are different in the two cases. To avoid this problem and provide a better head-to-head test, the two algorithms were compared using the same

PAIRCOIL-predicted regions as input, thus allowing SCORER 2.0 to classify the same sequences as MultiCoil's oligomeric-state-prediction function. This represents a better real-world scenario where a user might not know where coiled-coil regions lie in their protein of interest. The full protein sequences of our pristine dataset were submitted to the MultiCoil web server with parameters for which PAIRCOIL correctly recognized the majority of our coiled-coil test set. For a window length of 21 and a detection cutoff of 0.01, PAIRCOIL successfully predicted $77/133$ dimeric and $29/33$ trimeric coiled-coil regions. A coiled-coil region prediction by PAIRCOIL was considered *successful* if it encompassed at least 11 SOCKET-assigned coiled-coil residues. This ensures that the predicted coiled-coil regions are structurally verified coiled coils and are not a false-positive assigned by PAIRCOIL. The successful coiled-coil region assignments were then submitted to SCORER 2.0 and subsequently compared to the MultiCoil predictions.

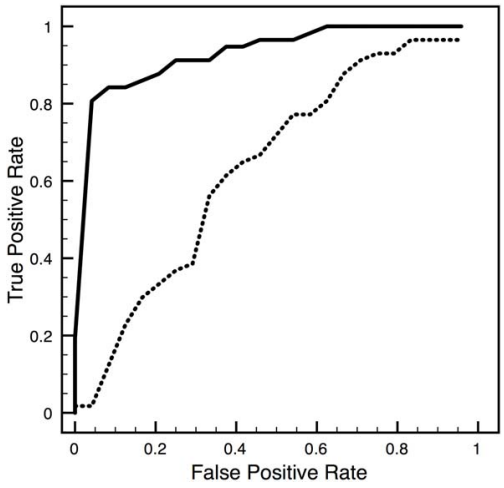

Fig. 3: ROC curves for SCORER 2.0 and MultiCoil when used to classify the oligomeric state of coiled coils using the PAIRCOIL-predicted regions. Solid line SCORER 2.0; dotted line MultiCoil. AUC for SCORER 2.0: 0.89, AUC for MultiCoil: 0.59. Test set comprised 72 dimeric sequences and 29 trimeric sequences.

Again, the AUC values and ROC curves in Figure 3 show that SCORER 2.0 achieved a higher discrimination rate than MultiCoil (0.89 vs 0.59, respectively). Table 1 compares the performance of SCORER 2.0 and MultiCoil across a wide range of PAIRCOIL parameters. For each of these parameter-sets, AUC scores and the fraction of correctly assigned dimers and trimers were used as a metric of how well the two algorithms performed in assigning oligomeric state. From these data two major trends are apparent: first, there is a correlation between the confidence of PAIRCOIL predictions and the accuracy of the corresponding MultiCoil oligomeric-state prediction. Whilst this is also true for SCORER 2.0, the effect is much greater for MultiCoil. We found that the median Spearman's rank correlation coefficient between AUC values and PAIRCOIL cut-off was 0.98 for MultiCoil (*i.e.*, a strong positive correlation), while a value of -0.27 is found for SCORER 2.0 (a weak negative correlation). As an example, for coiled coils 14 amino acids and longer using a window length of 21 and a PAIRCOIL cut-off of 0.01, MultiCoil achieved an AUC score of 0.59. When the PAIRCOIL cut-off was increased to 0.90, the equivalent score is 0.88. For those same PAIRCOIL parameters, the AUC values obtained when using SCORER 2.0 were 0.89 and 0.85. A closer look at the predictions reveals that MultiCoil performs very well on long, parallel dimers, but fails to replicate this for shorter dimers and trimers, suggesting it is tuned to output safe predictions, an observation that has been made by others (Gruber *et al.*, 2006).

In summary, SCORER 2.0 shows a sustained strong discrimination rate across a diverse range of coiled-coil sequences, while MultiCoil performs best for a more restricted set of long coiled-coil dimers. We suggest that this is a consequence of the redundancy in the MultiCoil training set, reflecting the availability of data at the time this software was released, rather than a flaw with the MultiCoil algorithm itself. Generally, SCORER 2.0 outperforms MultiCoil as an oligomeric state predictor, both in terms of AUC scores and correct predictions (Table 1).

3.5 Web-based interface for SCORER 2.0

An online resource has been constructed as an interface for the SCORER 2.0 algorithm at http://coiledcoils.chm.bris.ac.uk/Scorer. The SCORER 2.0 web server, source code and training set is freely available for academic users. Two options are made available for the user:

- A full protein sequence can be submitted as input. It is first processed by MARCOIL, where it is left to the user to freely choose a MARCOIL coiled-coil probability threshold (default is 50%). The SCORER 2.0 algorithm is then run on these MARCOIL-predicted coiled-coil regions.
- A coiled-coil sequence with assigned heptad register can also be submitted as input. In this case, SCORER 2.0 is run on the sequence immediately. It should be noted that the SCORER 2.0 algorithm also allows non-canonical coiled-coil sequences to be submitted as input, *i.e.*, those containing non-heptad repeats, although we emphasize that SCORER 2.0 was trained only on canonical coiled-coil sequences.

4 CONCLUSION

By retraining and revising the SCORER algorithm, a coiled-coil classifier written in 1995 (Woolfson and Alber, 1995), we have successfully predicted the oligomeric state of a range of dimeric and trimeric coiled-coil sequences with experimentally verified 3D structures. In nearly all cases, SCORER 2.0 offers improvement over the current standard in the field, MultiCoil. MultiCoil is good when classifying strongly defined coiled-coil sequences, but performs less well in other cases. We propose that this is most likely linked to redundancy in the MultiCoil training set reinforcing the redundancies found in the front-end PAIRCOIL algorithm. On the other hand, SCORER 2.0 was found to accurately distinguish between dimeric and trimeric coiled coils across the whole range of coiled-coil sequences used.

We propose that coiled-coil oligomeric state prediction is currently limited by two factors, (1) the accuracy of the coiled-coil-region prediction software used as the front-end, and (2) the number of oligomeric states included in the prediction, as coiled coil can be

PAIRCOIL PARAMETERS			TEST SET SIZE		MULTICOIL		SCORER 2.0	
cc length	window	threshold	Dimers	Trimers	AUC	Correctly assigned (Dimer/Trimer)	AUC	Correctly assigned (Dimer/Trimer)
≥ 14	21	0.01	77	29	0.59	0.44 / 0.69	0.89	0.86 / 0.72
		0.10	53	21	0.64	0.68 / 0.48	0.85	0.85 / 0.57
		0.50	32	11	0.79	0.94 / 0.45	0.87	0.94 / 0.73
		0.90	12	7	0.88	0.92 / 0.43	0.85	0.83 / 0.57
	28	0.01	70	29	0.67	0.41 / 0.86	0.90	0.87 / 0.69
		0.10	54	19	0.68	0.55 / 0.74	0.92	0.89 / 0.74
		0.50	36	11	0.71	0.86 / 0.54	0.91	0.86 / 0.73
		0.90	23	8	0.79	0.96 / 0.50	0.91	0.91 / 0.62
≥ 21	21	0.01	57	24	0.63	0.51 / 0.67	0.92	0.84 / 0.83
		0.10	42	18	0.66	0.73 / 0.50	0.90	0.86 / 0.67
		0.50	24	9	0.80	1.00 / 0.44	0.89	0.92 / 0.89
		0.90	11	7	0.93	1.00 / 0.43	0.84	0.82 / 0.57
	28	0.01	55	24	0.67	0.45 / 0.83	0.91	0.87 / 0.75
		0.10	44	17	0.69	0.59 / 0.70	0.92	0.89 / 0.76
		0.50	31	11	0.73	0.87 / 0.54	0.91	0.87 / 0.73
		0.90	20	8	0.81	1.00 / 0.50	0.93	0.89 / 0.62
≥ 28	21	0.01	35	16	0.74	0.63 / 075	0.95	0.94 / 0.87
		0.10	28	10	0.78	0.78 / 0.50	0.88	0.86 / 0.70
		0.50	18	6	0.96	1.00 / 0.67	0.93	0.89 / 0.83
		0.90	10	4	0.99	1.00 / 0.75	0.85	0.80 / 0.75
	28	0.01	35	16	0.77	0.57 / 0.94	0.91	0.86 / 0.75
		0.10	28	12	0.84	0.71 / 0.83	0.90	0.86 / 0.75
		0.50	22	8	0.95	0.95 / 0.75	0.92	0.91 / 0.75
		0.90	17	5	0.98	1.00 / 0.80	0.93	0.88 / 0.80

Table 1. Comparison of SCORER 2.0 and MultiCoil performance across a range of PAIRCOIL parameters. The PAIRCOIL parameters that were varied were the input coiled-coil sequence length (cc length), the PAIRCOIL window size (window) and the PAIRCOIL decision threshold (threshold). The test set obtained for each combination of these PAIRCOIL parameters was submitted to SCORER 2.0 for oligomeric state prediction (see text for details).

found in higher-order and more-complex oligomer states, as well as being parallel or anti-parallel (Walshaw and Woolfson, 2001; Lupas and Gruber, 2005; Moutevelis and Woolfson, 2009). We have used the PAIRCOIL front-end in this paper to ensure the results obtained from SCORER 2.0 and MultiCoil are comparable. However, other front-ends exist, MARCOIL (Delorenzi and Speed, 2002), PCOILS (Gruber *et al.*, 2006) and CCHMM-PROF (Bartoli *et al.*, 2009) have been shown to offer the best performances (Gruber *et al.*, 2006). At present, we use MARCOIL as a front-end to SCORER 2.0. Predicting oligomeric states beyond parallel dimers and trimers is limited mostly by the availability of sequence and structural data for other alternative oligomeric states. Homology based approaches such as SPIRICOIL (Rackham *et al.*, 2010) improve upon this by providing coiled-coil oligomeric state annotation as part of a Hidden Markov model used to classify whole proteins into families, but cannot be used to classify the oligomeric state of *de novo* coiled-coil sequences; *i.e.*, those without structurally defined precedents. Still, this kind of method may provide enough data of a high enough confidence to train algorithms such as SCORER 2.0 to predict between multiple oligomeric states, rather than just de novo dimeric and trimeric coiled coils. SCORER 2.0 shows little discrimination between the next two biggest classes of coiled-coil architecture — parallel tetramers and antiparallel dimers — when forced to assign an oligomeric state (Fig. S3, Supporting Information). We see the development of multi-state predictors to be the next logical step in coiled-coil structure analysis and prediction.

SCORER 2.0 is publicly and freely available via the world-wide web at http://coiledcoils.chm.bris.ac.uk/Scorer and can be used as stand-alone software for known coiled-coil regions, or in conjunction with MARCOIL, for coiled-coil region detection and oligomeric state assignment

ACKNOWLEDGEMENT

The authors would like to thank Dr. Beth Bromley and Dr. Gail Bartlett, and members of the Woolfson lab for several useful discussions. The authors would also like to acknowledge Dr. Mauro Delorenzi for allowing the free use of MARCOIL on the SCORER 2.0 web server.

Funding: CTA is funded by the BBSRC, TLV is funded by the EPSRC.

REFERENCES

Acharya, A., Rishi, V., and Vinson, C. (2006). Stability of 100 Homo and Heterotypic Coiled-Coil a-a' Pairs for Ten Amino Acids (A, L, I, V, N, K, S, T, E, and R). *Biochemistry*, **45**(38), 11324–11332.

Bartoli, L., Fariselli, P., Krogh, A., and Casadio, R. (2009). CCHMM_PROF: a HMM-based coiled-coil predictor with evolutionary information. *Bioinformatics*, **25**(21), 2757–2763.

Berger, B., Wilson, D. B., Wolf, E., Tonchev, T., Milla, M., and Kim, P. S. (1995). Predicting coiled coils by use of pairwise residue correlations. *Proc. Natl. Acad. Sci. U.S.A.*, **92**(18), 8259–8263.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Res*, **28**(1), 235–242.

Bromley, E. H. C., Sessions, R. B., Thomson, A. R., and Woolfson, D. N. (2009). Designed α-helical tectons for constructing multicomponent synthetic biological systems. *J Am Chem Soc*, **131**(3), 928–930.

Crick, F. H. C. (1953). The packing of α-helices - simple coiled coils. *Acta Crystallographica*, **6**(8-9), 689–697.

Delorenzi, M. and Speed, T. (2002). An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics*, **18**(4), 617–625.

Fariselli, P., Molinini, D., Casadio, R., and Krogh, A. (2007). Prediction of structurally-determined coiled-coil domains with hidden Markov models. In *Lect Notes Comput Sc*, volume 4414, pages 292–302.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recogn Lett*, **27**(8), 861–874.

Geman, S., Bienenstock, E., and Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural Comput*, **4**, 1–58.

Gough, J., Karplus, K., Hughey, R., and Chothia, C. (2001). Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J Mol Biol*, **313**(4), 903–19.

Gruber, M., Söding, J., and Lupas, A. N. (2006). Comparative analysis of coiled-coil prediction methods. *J Struct Biol*, **155**(2), 140–145.

Hadley, E. B., Testa, O. D., Woolfson, D. N., and Gellman, S. H. (2008). Preferred side-chain costellations at antiparallel coiled-coil interfaces. *Proc. Natl. Acad. Sci. U. S. A.*, **105**, 530–535.

Krylov, D., Barchi, J., and Vinson, C. (1998). Inter-helical interactions in the leucine zipper coiled coil dimer: pH and salt dependence of coupling energy between charged amino acids. *J Mol Biol*, **279**(4), 959–972.

Larkin, M. A., Blackshields, G., Brown, N. P., Chenna, R., McGettigan, P. A., McWilliam, H., Valentin, F., Wallace, I. M., Wilm, A., Lopez, R., Thompson, J. D., Gibson, T. J., and Higgins, D. G. (2007). Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**(21), 2947.

Li, W. Z. and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**(13), 1658–1659.

Lupas, A. N. and Gruber, M. (2005). The structure of α-helical coiled coils. *Adv Protein Chem*, **70**, 37–78.

Lupas, A. N., Van Dyke, M., and Stock, J. (1991). Predicting coiled coils from protein sequences. *Science*, **252**(5009), 1162–1164.

Mason, J. M., Schmitz, M. A., Müller, K. M., and Arndt, K. M. (2006). Semirational design of Jun-Fos coiled coils with increased affinity: Universal implications for leucine zipper prediction and design. *Proc. Natl. Acad. Sci. U. S. A.*, **103**(24), 8989.

McDonnell, A. V., Jiang, T., Keating, A. E., and Berger, B. (2006). Paircoil2: improved prediction of coiled coils from sequence. *Bioinformatics*, **22**(3), 356–358.

Moutevelis, E. and Woolfson, D. N. (2009). A periodic table of coiled-coil protein structures. *J Mol Biol*, **385**(3), 726–732.

Newman, J. R. S. and Keating, A. E. (2003). Comprehensive identification of human bZIP interactions with coiled-coil arrays. *Science*, **300**(5628), 2097.

Parry, D. A. D. (1982). Coiled-coils in α-helix-containing proteins - analysis of the residue types within the heptad repeat and the use of these data in the prediction of coiled-coils in other proteins. *Bioscience Reports*, **2**(12), 1017–1024.

Rackham, O. J. L., Madera, M., Armstrong, C. T., Vincent, T. L., Woolfson, D. N., and Gough, J. (2010). The Evolution and Structure Prediction of Coiled Coils across All Genomes. *J Mol Biol*, **403**, 480–493.

Reinke, A. W., Grant, R. A., and Keating, A. E. (2010). A Synthetic Coiled-Coil Interactome Provides Heterospecific Modules for Molecular Engineering. *J Am Chem Soc*, **132**(17), 6025–6031.

Rice, P., Longden, I., and Bleasby, A. (2000). EMBOSS: the European Molecular Biology Open Software Suite. *Trends in Genetics*, **16**(6), 276–277.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics*, **21**(20), 3940–3941.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, **147**(1), 195–197.

Steinkruger, J. D., Woolfson, D. N., and Gellman, S. H. (2010). Side-Chain Pairing Preferences in the Parallel Coiled-Coil Dimer Motif: Insight on Ion Pairing between Core and Flanking Sites. *J Am Chem Soc*, **132**(22), 7586–7588.

Tanizawa, H., Chimire, G. D., and Mitaku, S. (2008). A high performance prediction system of coiled coil domains containing heptad breaks: SOSUIcoil. *Chem-Bio Informatics*, **8**(3), 96–111.

Testa, O. D., Moutevelis, E., and Woolfson, D. N. (2009). CC+: a relational database of coiled-coil structures. *Nucleic Acids Res*, **37**(Database issue), D315–22.

Walshaw, J. and Woolfson, D. N. (2001). SOCKET: a program for identifying and analysing coiled-coil motifs within protein structures. *J Mol Biol*, **307**(5), 1427–1450.

Wolf, E., Kim, P. S., and Berger, B. (1997). MultiCoil: a program for predicting two- and three-stranded coiled coils. *Protein Sci*, **6**(6), 1179–1189.

Woolfson, D. N. and Alber, T. (1995). Predicting oligomerization states of coiled coils. *Protein Sci*, **4**(8), 1596–1607.