**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 recognized 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 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 parallel dimers and trimers.

**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:** coiledcoils-scorer@sympa.bristol.ac.uk; d.n.woolfson@bristol.ac.uk

**Supplementary information:** Supplementary data are available at Bioinformatics online.

Received on December 16, 2010; revised on April 12, 2011; accepted on May 4, 2011

**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 d 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 antiparallel 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?

Here, we focus on the second problem, and specifically on oligomer-state prediction. However, the first problem of locating coiled-coil

© The Author 2011. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com
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) utilizes 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 utilizing 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. SOSUcoil
(Tantawra et al., 2008) uses physico-chemical parameters—such
as predicted sequence amphipathicity—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
SUPERFAMILY 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 truly \emph{ab initio} coiled-coil prediction algorithms, MARCOIL
and PCOILS are thought to perform best (Gruber et al., 2006),
although SOSUcoil, CCHMM and CCHMM-PROF are yet to be
independently benchmarked.

Three algorithms exist to tackle the architecture problem:
SCORER (Woolfson and Alber, 1995), MultiCoil (Welt 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
two 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 \emph{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 stabilize dimeric assemblies of coiled coils
have been analysed using experimental and bioinformatic methods
(Acharya et al., 2006; Hadley et al., 2008; Krylov et al., 1998;
Mason et al., 2006; Newman and Keating, 2003; Reinke et al.,
2010; Steinrueger 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 (Lupas and Gruber, 2005; Moutevelis
and Woolfson, 2009; Walshaw and Woolfson, 2001) and the current
limits of homology based coiled-coil prediction such as SPIRICOIL,
one of our focuses has been on improving \emph{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 ∼ 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 1.0 algorithm. SCORER 2.0 uses
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 seven 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 with 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 cut-off 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 \emph{pristine dataset}. The identity threshold of 25–30%, often used for calling 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 Supplementary Material). The pristine dataset comprised 133 dimeric
and 33 trimeric coiled-coil sequences. Position-specific scoring matrices,
PSSMs (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 \times 7 tables. Each element of these
tables was denoted \( P_{ij} \), where \( i \) can take the values 2 or 3 to
denote the dimer and trimer PSSMs, 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 \( T_{ijr} \).
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 of the data sets are given at http://coiledcoils.chem.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:

\[
\text{Score} = \sum_{i=1}^{r} \log \frac{\text{PROF}_{i,a}}{\text{PROF}_{i,b}}
\]

where \( A = \{a_1, \ldots, a_\ell \} \) and \( R = \{r_1, \ldots, r_\ell \} \) represents the amino acid residues and associated register positions of the test coiled-coil sequence \( S \) with residues numbered \( \ell = 1, 2, \ldots, \ell \) and observed oligomeric state \( O = (2, 3) \).

By selecting the terms in the dimer and trimer PSSMs 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 versus 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 versus trimer relative to the user’s 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\(_{i,a}\), for example, if a particular residue \( a \) is rare for a particular oligomeric type \( o \) and register \( r \) then PROF\(_{i,a}\) 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:

\[
\text{Score}_{a} = \log \left( \frac{\text{TOT}_{a} (2^{\alpha} + 2^{\beta} + 1)}{\text{TOT}_{a} (2^{\alpha} + 2^{\beta} + 1)} \prod_{r} \text{PSSM}_{a,r} + \delta^{(\alpha + \beta)} \right) - \log \left( \prod_{r} \text{PSSM}_{a,r} + \delta^{(\alpha + \beta)} \right)
\]

where \( x_{a,r} \) is the number of \( a \) residues at register \( r \) in the test sequence, \( y_{r} = \sum x_{a,r} \) is the total number of amino acids at register \( r \) in the test sequence, and \( m^{\alpha} \) stands for the rising factorial symbol \( m \cdot (m+1) \cdot (m+2) \cdot \ldots \cdot (m+\alpha-1) \).

The constant \( \delta \) 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 \( \delta \) 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 recognized 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 publicly 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:

\[
S_{a} = \frac{D_{a} - T_{a}}{C_{a}}
\]

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 TPR is the probability of correctly classifying a true instance and is defined as:

\[
\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}
\]

The FPR is the probability of assigning a false instance as true, and is defined as:

\[
\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}
\]

Here, TP, FN, FP, and TN represent counts of true positives, false negatives, false positives and true negatives, respectively. Data points are plotted in the 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 datasets. In the hypothetical case of the perfect separation of two datasets, 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 trade-off (Geran 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% (Supplementary Fig. S1).

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 dimeric and trimeric 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 the sequences, yielding an AUC of 0.77. When restricted to only those sequences that the original SCORER algorithm assigned, SCORER 2.0 achieved an AUC of 0.76 (Supplementary Fig. S2). 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 Section 2) 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 versus 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 oligomeric state 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.

3.4 SCORER 2.0 versus 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
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, well on long, parallel dimers, but fails to replicate this for shorter ones. MultiCoil discriminates between different oligomeric states of coiled coils with a strong positive correlation, while a value of 0.89, the median Spearman’s rank correlation coefficient between AUC for MultiCoil: 0.89 and 0.59.

When the PAIRCOIL cutoff was increased to 0.59, the effect is much greater for MultiCoil. We found that SCORER 2.0 achieved a higher discrimination rate than MultiCoil (0.89 versus 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 compared with the MultiCoil predictions.

Again, the AUC values and ROC curves in Figure 3 show that SCORER 2.0 achieved a higher discrimination rate than MultiCoil (0.89 versus 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 compared with the MultiCoil predictions.

A full protein sequence can be submitted as input. It is first processed by PAIRCOIL, 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: (i) the accuracy of the coiled-coil region prediction software used as the front-end and (ii) the number of oligomeric states included in the prediction, as coiled coil can be found in higher order and more-complex oligomer states, as well as being parallel or anti-parallel (Lupas and Gruber, 2005; Moult and Woolfson, 2009; Walshaw and Woolfson, 2001). We have used the PAIRCOIL front-end in this article 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...
Table 1. Comparison of SCORER 2.0 and MultiCoil performance across a range of PAIRCOIL parameters

<table>
<thead>
<tr>
<th>PAIRCOIL Parameters</th>
<th>Test Set Size</th>
<th>Multicoil</th>
<th>SCORER 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Dimers</td>
<td>Trimers</td>
</tr>
<tr>
<td>cc length</td>
<td>Window</td>
<td>Threshold</td>
<td>(dimer/trimer)</td>
</tr>
<tr>
<td>≥ 14</td>
<td>21</td>
<td>0.01</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.15</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.70</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.90</td>
<td>12</td>
</tr>
<tr>
<td>28</td>
<td>21</td>
<td>0.01</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.60</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.70</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.90</td>
<td>23</td>
</tr>
<tr>
<td>≥ 21</td>
<td>21</td>
<td>0.01</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.60</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.70</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.90</td>
<td>11</td>
</tr>
<tr>
<td>28</td>
<td>21</td>
<td>0.01</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.60</td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.70</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.90</td>
<td>20</td>
</tr>
<tr>
<td>≥ 28</td>
<td>21</td>
<td>0.01</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.60</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.70</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.90</td>
<td>10</td>
</tr>
<tr>
<td>28</td>
<td>21</td>
<td>0.01</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>0.10</td>
<td>0.60</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>0.70</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>0.90</td>
<td>17</td>
</tr>
</tbody>
</table>

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 text set obtained for each combination of these PAIRCOIL parameters was submitted to SCORER 2.0 for oligomeric state prediction (see text for details).

ACKNOWLEDGEMENTS

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: BBSRC (to C.T.A.); EPSRC (to T.L.V.).

Conflict of Interest: none declared.

REFERENCES


C.T.Armstrong et al.


