COMP0082 Coursework V2.02 - Bioinformatics, 2025, pp. 1–5

doi: DOI
Paper

PAPER

Predicting the subcellular location of proteins

Student Number: 24237262[1,*]

[1]Computer Science, University College London, 66-72 Gower St, WC1E 6EA, London, United Kingdom

Abstract

Over the last decade, the complete sequence has been determined for thousands of genomes. This has created the need for fully automated methods to analyse the vast amount of sequence data now available. The assignment of a function for a given protein has proved to be difficult where no clear homology to proteins of known function exists. Knowing the subcellular location of a protein might give some clue as to its possible function, making an automated method that assigns proteins to a certain subcellular location a useful tool for analysis. This work presents an approach for predicting the subcellular location (nuclear, mitochondrial, cytosolic, secreted/extracellular or other) of non-homologous proteins.This method extracts various structural, physicochemical, and compositional features from protein sequences, including sequence properties, amino acid frequencies, and nucleotide content, and performs classification using a Random Forest (RF). Results show that the RF classifier is able to effectively exploit the engineered features, achieving an accuracy of 65% on an independent test set consisting of 1,782 non-homologous amino acid sequences. The source code is freely available online on `https://github.com/iamgiulyUCL/COMP0082-Bioinformatics`

Introduction

Eukaryotic cells have evolved ways to partition off different functions to various locations in the cell. Different organelles play different roles in the cell, and proteins within them perform a vast number of functions that enable the well-functioning of the organism, including DNA replication, catalyzing metabolic reactions, and transporting molecules from one location to another. Understanding the role of proteins based on their amino acid residues has centered the effort of the scientific community for several years, and determining where these proteins reside in the cell might be a critical milestone towards this goal. Numerous efforts have been made to develop methods for predicting protein subcellular location based on the amino acid sequence information. TargetP (Emanuelsson et al. [2000]) assigns protein location with a feed-forward neural network, and it is based on the predicted presence of specific N-terminal presequences: chloroplast transit peptide (cTP), mitochondrial targeting peptide (mTP), or secretory pathway signal peptide (SP). WoLF PSORT (Horton et al. [2007]) converts protein amino acid sequences into numerical localization features based on sorting signals, amino acid composition, and functional motifs such as DNA-binding motifs, and uses a K-nearest neighbor classifier for prediction. Several other methods exist, such as ProLoc-GO (Huang et al. [2008]) or KnowPredsite (Lin et al. [2009]) is less effective when dealing with non-homologous proteins because it relies on ontologies or knowledge graphs. It is well known that most proteins in eukaryotic cells are synthesized in the cytosol, and many need to be further sorted to cellular organelles. This process usually relies on specific signal peptides located in the N-terminal, such as the mitochondrial targeting peptide (mtTP) or the secretory pathway signal peptide (SP). These signals are recognized by a translocation machinery, and the proteins are delivered to the corresponding organelle. Once the protein arrives at the destination, the signal peptide is typically removed by a signal peptidase. In this study, several global and local features are extracted from the amino acid sequence to capture meaningful patterns useful for predicting subcellular locations. To this end, this work focuses on extracting both global and local features from the amino acid sequences using Random Forest (RF).A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. In addition, RF has been widely used in bioinformatics because the data complexity is always rising, and as a nonparametric model, random forest provides a unique combination of prediction accuracy and model interpretability (Qi [2012]). In summary, this study investigates how a rich set of physicochemical and sequence-based features, combined with machine learning model Random Forests, can enhance our understanding of protein subcellular localization, ultimately contributing to more accurate predictive models.

Methods

Data and preprocessing

The dataset included 9,460 amino acid sequences, each labeled with a subcellular location (cytosolic, nuclear, mitochondrial, or secreted/extracellular) and encoded in FASTA format, plus "Other" which contains examples of prokaryotic proteins that sometimes contaminate samples during sequencing. Additionally, a separate set of 20 unlabeled sequences was provided to retrospectively assess generalization performance.

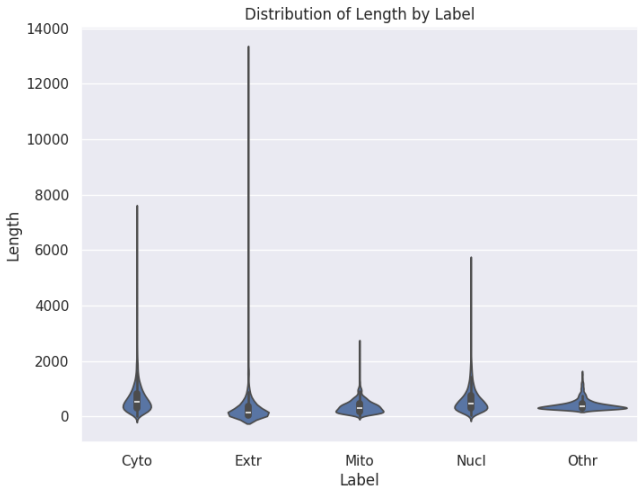

Fig. 1. Distribution of length for each subcellular location

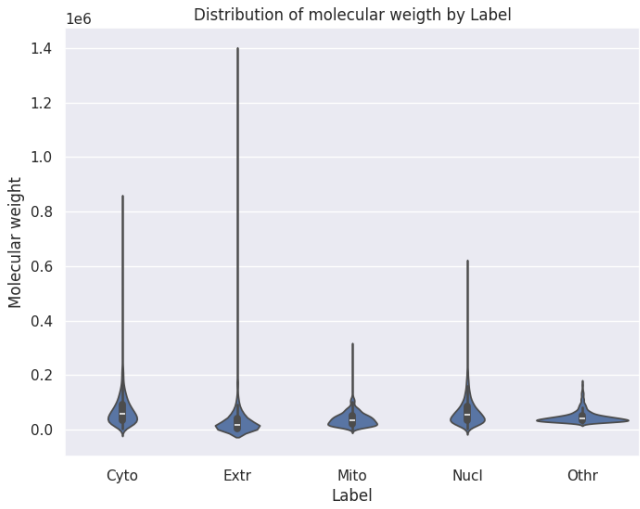

Fig. 2. Distribution of molecular weigth for each subcellular location

The labeled data was divided into training (7,568 sequences) and test (1,892 sequences) sets using stratified random sampling. The sequences contained no gaps or translation stop signals but occasionally included the "X", "B" and "U" (Selenocysteine) codes, representing an unknown amino acid in FASTA format. All of them were ignored because they were in the 0.6% of the samples.

Feature engineering

The input features were designed to capture key properties of the amino acid sequences using the publicly available Biopython library (Cock et al. [2009]). These features include: (1) sequence length, (2) molecular weight, (3) isoelectric point (4) charge distribution at pH 7.4, (5) aromaticity, (6) flexibility, (6) secondary structure fraction, (7) hydropathy index, (8) charge-to-isoelectric point ratio, and (9) molecular weight per sequence length. Additionally, amino acid frequency features were calculated for each amino acid in the sequence, and GC content and AT content were included (Qing-Bin Gao [2005]). Figures 1, 2, 3 and 4 show the discriminative power of some engineered features by themselves.

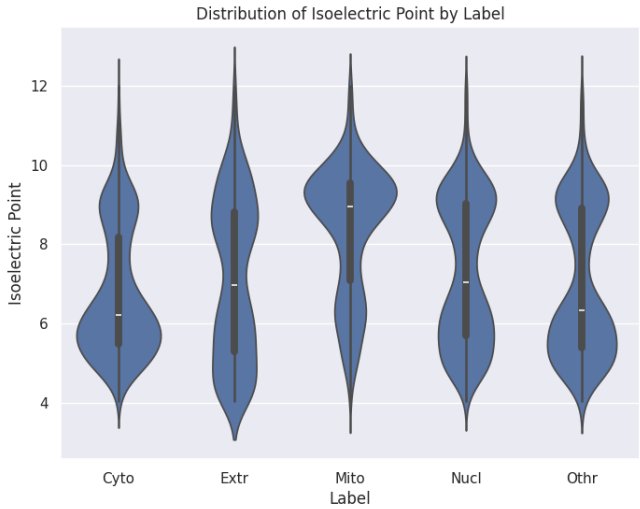

Fig. 3. Distribution of isoelectric point for each subcellular location

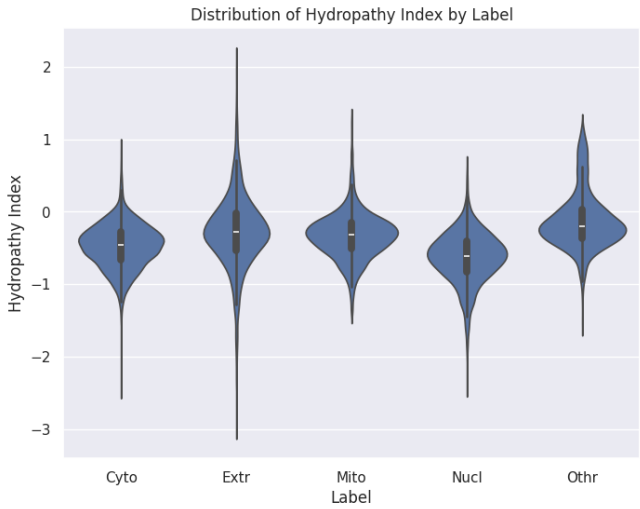

Fig. 4. Distribution of hydropathy index for each subcellular location

Further analysis was conducted through an inspection of feature correlations with the aim of refining the input variables for classification. In particular, a special focus was given to highly correlated features, as their presence in the model can introduce redundancy and obscure the true contribution of individual predictors. There is a direct comparison of molecular_weight and sequence_length which reveals a perfect correlation (r = 1.00), indicating that these two features carry equivalent information. Given that sequence length is a fundamental biological property, while molecular weight is derived from it, the latter was removed to prevent redundant contributions to the model. Similarly, the relationship between charge_to_isoelectric_point_ratio and charge_distribution (r = 0.97) suggested that these features capture similar charge-related properties. To ensure clarity in feature interpretation, only charge_distribution was retained, as it provides a more direct measure of charge variability across the protein sequence. In contrast, moderately correlated features such as at_content and A_freq (r = 0.86) were both preserved, as they describe distinct aspects of nucleotide composition that may contribute meaningfully to classification. Finally, a strong inverse correlation was observed between mean_flex and hydropathy_index (r = -0.85), reflecting the well-established relationship between protein flexibility and hydrophobicity. Since these features offer different biological insights, they were retained to allow the model to capture their joint effect on protein classification. Further inspection of the dataset revealed an imbalance in class distribution, with certain subcellular locations being underrepresented compared to others. Class imbalance can introduce biases in classification models, leading to a tendency to favor majority classes while misclassifying minority classes. To mitigate this issue, a stratified train-test split was employed, ensuring that the relative proportions of each class were maintained in both the training and testing sets. Additionally, stratified k-fold cross-validation was utilized during model training. This approach ensures that each fold maintains the same class distribution as the original dataset, preventing the classifier from being skewed toward dominant classes while allowing for a more robust evaluation of predictive performance. By incorporating these techniques, we aim to balance the learning process across all classes and improve the model's ability to recognize minority class instances. The dataset comprised 35 distinct features. Further preprocessing was necessary to ensure all features were within a reasonable range. For example, some features have missing values or extreme disparities. To address this, missing values were imputed using the mean strategy, and the features were normalized by removing the mean and scaling to unit variance. These preprocessing steps were incorporated within the hyperparameter tuning process using GridSearchCV (sci [2025]).

Model design

Random Forest (RF) (Breiman [2001]) classifiers are generally less affected by multicollinearity, as their tree-based structure allows for the independent selection of features across different splits. This makes them particularly suitable when dealing with highly correlated features. (Ceh and Kilibarda [2018]). The model design was based on a RF, that is inherently designed to handle multiclass classification. The publicly available scikit-learn software was used to train the model (Pedregosa et al. [2018]). Random forest is a classification and regression algorithm based on the bagging and random subspace methods. The idea of bagging is to construct an ensemble of learners, each trained on a bootstrap sample (Db) obtained from the original dataset (D) using the following sampling procedure: given a D with N examples, one creates a Db by randomly choosing k examples from D with replacement (after selecting an example, it is immediately returned to D and can be selected again). After removing duplicates, if N is large and k = N, it is expected that Db contains approximately two-thirds of examples from D. The prediction of the ensemble is constructed from the separate decisions by majority voting (classification) or averaging (regression). It has been shown that bagging can reduce the variance in the final model when compared to the base models and can also avoid overfitting. In this work the classification version has been used, where trees in the forest use the best split strategy, i.e. equivalent to passing splitter="best" to the underlying DecisionTreeClassifier. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree. In this work the parameter has left as default.

There are several mathematical principles that underlie its design, including: (1) Ensemble Learning Theory: Random Forest belongs to the family of ensemble methods, which combine multiple models to improve performance. The key mathematical principle behind ensemble learning is the bias-variance tradeoff, (2) the Law of Large Numbers which explains how, as the number of trees in the Random Forest increases, the predictions of the individual trees become less important and the model's output becomes more stable and reliable. This is similar to the Law of Large Numbers applied to independent and identically distributed random variables, which explains that the value tends to the expected value as the number of variables increases. In Random Forest, the more trees you add, the more the ensemble's prediction converges to the true underlying function, reducing overfitting, (3) the correlation between Trees that explains a critical mathematical insight behind Random Forest's success. More specifically that while the trees are not completely independent, better (lower generalization error) random forests have lower correlation between classifiers and higher strength. The randomness used in tree construction has to aim for low correlation $\overline{rho}$ while maintaining reasonable strength (Breiman [2001]). The performance of RF classifiers was evaluated via stratified 4-fold cross-validation, and the model's hyperparameters (such as the number of estimator, the maximum depth of the trees and the minimum number of sampled) were tuned to optimize the classification accuracy.

Result and discussion

The final tuned classifier, with GridSearchCV, includes bootstrap set as True. The model was trained on a set with 7568 sequences, and tested on a set with 1892 sequences, split randomly in a stratified manner. The hyperparameters of the model (n_estimators = 300, max_depth = None, min_samples_split = 2) were determined using stratified 4-fold cross-validation. The overall accuracy of the final predictor on the test set was 65%. Table 1 summarize the prediction performance of the RF in terms of Accuracy, Precision, Recall, F1-score and Area Under the Curve (AUC). The latter is computed using the one-vs-one strategy, and then the final result is weighted. Results show an overall good performance for all metrics.

Figure 5 displays the Receiver Operating Characteristic (ROC) curves for five different subcellular locations: Cytoplasm (Cyto), Extracellular (Extr), Mitochondria (Mito), Nucleus (Nucl), and Other (Othr). The ROC curve illustrates the trade-off between the true positive rate (sensitivity) and the false positive rate for a multi-class classification model. The Area Under the Curve (AUC) is used as a performance metric, where a higher AUC value indicates better model performance. Among the classes, the model performs best in identifying the 'Other' category with an AUC of 0.97, followed closely by 'Extracellular' with an AUC of 0.95. Both 'Mitochondria' and 'Nucleus' share a strong AUC of 0.86. The 'Cytoplasm' class shows relatively lower performance with an AUC of 0.79, suggesting more difficulty in distinguishing this class compared to the others. The dotted line corresponds to the random performance. Overall, the model demonstrates robust classification ability for most subcellular locations, with especially high sensitivity and specificity for the 'Other' and 'Extracellular' categories. Figure 6 presents the confusion matrix for the classification of subcellular locations, where the rows represent the true labels and the columns denote the predicted labels. Diagonal values indicate correct predictions, while off-diagonal values represent misclassification. The model demonstrates strong performance in classifying Nucleus (Nucl) and Other (Othr) categories, with 383 and 351 correct predictions respectively, which aligns with their high AUC scores in the ROC analysis. The Extracellular (Extr) class also shows solid performance with 179 correct predictions and relatively few misclassification. In contrast, Cytoplasm (Cyto) exhibits considerable confusion, particularly with Nucleus (159 misclassifications) and to a lesser extent with Other (38) and Mitochondria (28). This likely explains the lower AUC score observed for this class. Mitochondria (Mito) classification also faces challenges, as it is frequently misclassified across all other classes, notably with Cytoplasm and Nucleus, indicating overlap or ambiguity in features between these compartments. These findings suggest that while the model performs well overall, improvements could focus on reducing confusion between Cytoplasm, Mitochondria, and Nucleus, potentially through feature engineering or class-specific data augmentation. Figure 7 shows the 25 top features ranked by importance, extracted by the RF best estimator. Feature importance reflects how much each variable contributes to the model's predictive power. The most influential feature is sequence_length, followed by C_freq (Cysteine frequency) suggesting that basic sequence composition and general protein size are critical determinants of subcellular location. Flexibility-related features such as mean_flex, coil_fraction, and max_flex also rank highly, indicating that structural dynamics of proteins play a significant role in localization. Other notable features include hydropathy_index and isoelectric_point, which relate to the biochemical behavior of proteins in different cellular environments. Frequencies of specific amino acids (e.g., S_freq, K_freq, A_freq) also appear important, possibly reflecting motifs or targeting signals relevant to localization. Lower-ranked features such as charge_distribution and helix_fraction still contribute, but to a lesser extent. This hierarchy of features can guide future biological interpretation, feature selection, or refinement of the model. Finally, table X shows the predictions of the proposed method for the 20 blind sequences.

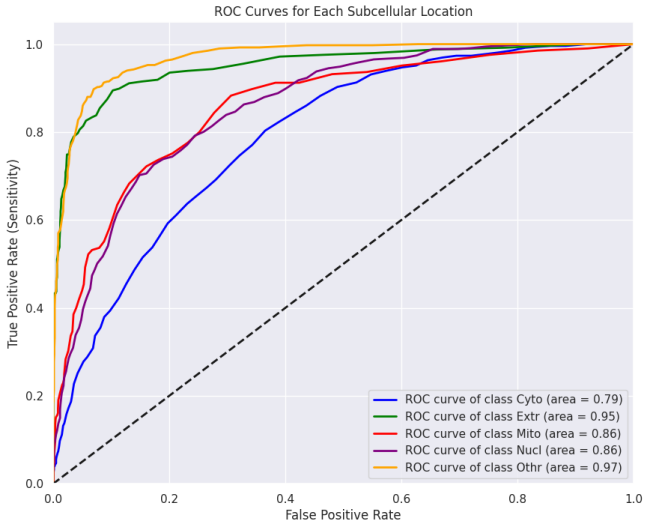

Fig. 5. The predictive performance shown as sensitivity versus false positive rate for each subcellular location. This plot was created from results obtained using the independent test set. The dotted line corresponds to the random performance.

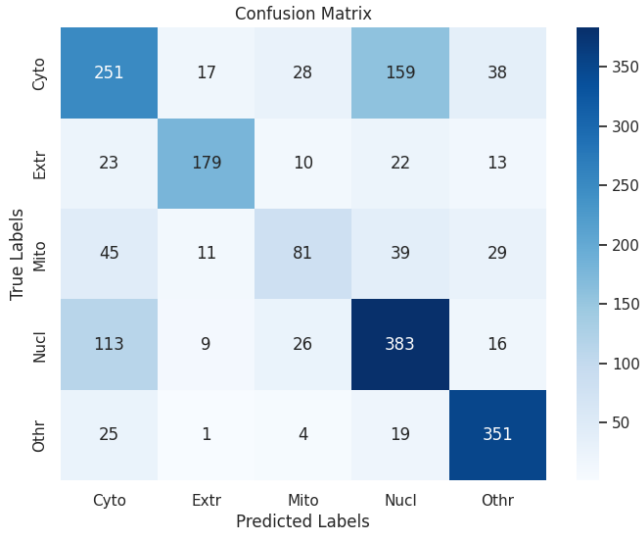

Fig. 6. Error analysis of the RF shown as a confusion matrix. The plot was constructed from results obtained using the independent test set. The true category and the predicted category are represented in the y-axis and x-axis, respectively.

Table 1. The prediction performance of the RF. This table was created from results obtained using the independent test set. The system obtains a good overall performance for each metric

Accuracy	Precision	Recall	F1-score	AUC
0.649926	0.660545	0.663319	0.659455	0.890312

Conclusion and future work

The aim of this work was to provide a method for predicting the subcellular location (nucleus, cytosol, mitochondrion, secretory or other) of eukaryotic proteins based on their amino acid sequence.The method presented in this work has dem

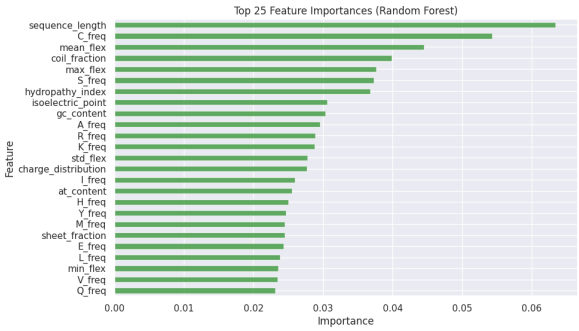

Fig. 7. Top 25 most important features for subcellular location classification based on Random Forest feature importance scores. Feature importance reflects each variable's contribution to the model's predictive performance.

onstrated a decent generalization performance on independent test set of sequences, but improvements can be done for unseen data. Predicting the subcellular location of eukaryotic proteins with RFs captured by the proposed set of features have been demonstrated to be informative of the subcellular location. Random Forests seem to be promising approach for predicting the subcellular location, mainly because of its known generalization ability. This set of features has significatively increased the ability of the model to discriminate nuclear and cytosolic proteins, but they are often not specific enough. Further approaches may benefit from attempting to capture richer patterns, potentially by matching profiles of signaling sequences or searching for known protein localization signals in manually curated databases such as LocSigDB (Negi S et al, 2015).

References

Gridsearchcv scikit-learn, 2025. URL `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`. Accessed: 2025-04-04.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001. doi: 10.1023/A:1010933404324.

M. Ceh and e. a. Kilibarda. Estimating the performance of random forest versus multiple regression for predicting prices of the apartments. *ISPRS International Journal of Geo-Information*, 7(5):168, 2018. doi: 10.3390/ijgi7050168.

P. J. A. Cock, C. B. A. Antao, Chang, Cox, et al. Biopython: Freely available python tools for computational molecular biology. *Bioinformatics*, 25(11):1422–1423, 2009. doi: 10.1093/bioinformatics/btp163.

O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne. Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. *Journal of Molecular Biology*, 300(4):1005–1016, 2000. doi: 10.1006/jmbi.2000.3903. URL `https://doi.org/10.1006/jmbi.2000.3903`.

P. Horton, K. J. Park, T. Obayashi, N. Fujita, H. Harada, C. J. Adams-Collier, and K. Nakai. Wolf psort: protein localization predictor. *Nucleic acids research*, 35(Web Server issue):W585–W587, 2007. doi: 10.1093/nar/gkm259. URL `https://doi.org/10.1093/nar/gkm259`.

W. L. Huang, C. W. Tung, S. W. Ho, S. F. Hwang, and S. Y. Ho. Proloc-go: utilizing informative gene ontology terms for sequence-based prediction of protein subcellular localization. *BMC bioinformatics*, 9:80, 2008. doi: 10.1186/1471-2105-9-80. URL `https://doi.org/10.1186/1471-2105-9-80`.

H. N. Lin, C. T. Chen, T. Y. Sung, S. Y. Ho, and W. L. Hsu. Protein subcellular localization prediction of eukaryotes using a knowledge-based approach. *BMC Bioinformatics*, 10(Suppl 15):S8, 2009. doi: 10.1186/1471-2105-10-S15-S8. URL `https://doi.org/10.1186/1471-2105-10-S15-S8`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python, 2018. URL `https://arxiv.org/abs/1201.0490`.

Y. Qi. Random forest for bioinformatics. *Ensemble machine learning: Methods and applications*, pages 307–323, 2012.

C. Y. Y.-H. D. Qing-Bin Gao, Zheng-Zhi Wang. Prediction of protein subcellular location using a combined feature of sequence. *FEBS Letters*, 579(16):3444–3448, 2005. doi: 10.1016/j.febslet.2005.05.021.