

Supplementary Methods

The UCYN-A transit peptide: a novel C-terminal targeting system
with distinctive biophysical substrate signatures

Contents

S1 Computational Environment

All analyses were performed using Python 3.12. Package management used uv (version 0.5.x). Key dependencies and versions:

Package	Version
BioPython	1.84
NumPy	2.0.x
pandas	2.2.x
scikit-learn	1.5.x
SciPy	1.14.x
matplotlib	3.9.x
seaborn	0.13.x
PyTorch	2.4.x
transformers	4.44.x
h5py	3.11.x
statsmodels	0.14.x

S2 Data Sources and Preprocessing

S2.1 uTP protein identification

We used the uTP HMM profile from Coale et al. to scan the *B. bigelowii* transcriptome using HMMER 3.4. Parameters:

- E-value threshold: 0.01
- Minimum bit score: 30.0
- HMM coverage requirement: match must start within first 50 positions of the profile

This identified 933 import candidates. The HMM hit position was used to define the boundary between mature domain and uTP region for each protein.

S2.2 Sequence datasets

Dataset	Count	Description
Import candidates	933	HMM-predicted uTP proteins
Experimentally validated	206	Enriched in UCYN-A proteomics
<i>B. bigelowii</i> proteome	44,363	Full predicted proteome

S3 uTP Sequence Organization

S3.1 Motif discovery

Motif discovery was performed using MEME Suite 5.5.5 on 206 experimentally validated uTP sequences. The C-terminal regions were extracted and filtered using Gblocks to remove poorly aligned positions. MEME parameters:

- Mode: zoops (zero or one occurrence per sequence)
- Number of motifs: 10
- Minimum motif width: 6
- Maximum motif width: 50
- Background model: order-0 Markov from input sequences

S3.2 Motif scanning

We scanned all 933 import candidates for the discovered motifs using MAST (Motif Alignment and Search Tool). Parameters:

- E-value threshold: 10.0 (permissive, to detect weak hits)
- Output: hit positions, p-values, and motif order

Motif patterns were classified by terminal motif identity:

- terminal_4: sequences ending with motif 4
- terminal_5: sequences ending with motif 5
- terminal_7: sequences ending with motif 7
- terminal_9: sequences ending with motif 9
- other: sequences with non-standard terminal motifs

S4 Structure Prediction and Analysis

S4.1 AlphaFold3 structure prediction

Structures were predicted using AlphaFold3 via the AlphaFold Server (<https://alphafoldserver.com>). We selected 47 proteins with high-confidence uTP regions (good C-terminal alignment in the multiple sequence alignment).

S4.2 Structural alignment

Structures were aligned using the CE (Combinatorial Extension) algorithm implemented in PyMOL. The C-terminal uTP regions (approximately 120 residues) were extracted and aligned to the structure with the longest C-terminal chain as reference.

Pairwise RMSD was computed using BioPython's Superimposer class on C α atoms.

S4.3 Consensus structure

The consensus structure was built as follows:

1. Identify reference structure (longest chain)

2. For each reference residue position, find spatially corresponding residues across all structures (within 3.0 Å of the reference position)
3. Compute consensus position as the mean of all corresponding C α coordinates
4. Record positional standard deviation as a measure of variance

S4.4 Secondary structure assignment

Secondary structure was assigned using DSSP algorithm criteria applied to backbone dihedral angles. Helices were defined as regions with ≥ 4 consecutive residues in helical conformation ($\phi \approx -60$, $\psi \approx -45$).

S5 Sequence Space Analysis

S5.1 Sequence embeddings

Protein sequences were encoded using the ProtT5-XL-UniRef50 model. For each sequence:

1. Tokenize sequence using the ProtT5 tokenizer
2. Pass through the encoder (no decoder)
3. Extract per-residue embeddings from the final hidden layer
4. Mean-pool across residue positions to obtain a single 1024-dimensional vector

Embeddings were computed on GPU (NVIDIA RTX PRO 6000) and cached in HDF5 format.

S5.2 Clustering methods

Four clustering methods were applied:

	Method	Implementation	Parameters
Hierarchical	scipy.cluster.hierarchy		k=4,5,6,7; linkage=average
Spectral	sklearn.cluster.SpectralClustering		k=4,5,6,7; affinity=rbf
K-means	sklearn.cluster.KMeans		k=4,5,6,7; n_init=10
DBSCAN	sklearn.cluster.DBSCAN		eps=0.5,1.0,2.0; min_samples=5

For hierarchical clustering, pairwise distances were computed as Jaccard distance on k-mer frequency vectors (k=4).

S5.3 Clustering evaluation

Cluster quality was assessed using:

- **Silhouette score:** Mean silhouette coefficient across all samples. Range $[-1, +1]$; values > 0.5 indicate strong cluster structure.
- **Adjusted Rand Index (ARI):** Agreement between two clusterings, adjusted for chance. Range $[-1, +1]$; values near 0 indicate random agreement.

S5.4 Null model comparison

To test whether uTP sequences show more structure than expected by chance:

1. For each real uTP sequence, generate a null sequence by randomly shuffling amino acids (preserving exact composition)
2. Compute ProtT5 embeddings for all null sequences
3. Apply k-means clustering ($k=4$) and compute silhouette score
4. Repeat for 100 independent null sequence sets
5. Compare observed silhouette score to null distribution using one-sided permutation test

Additional metrics computed: Hopkins statistic, mean pairwise distance, distance variance.

S6 Mature Domain Classification

S6.1 Mature domain extraction

For each uTP protein, the mature domain was defined as the sequence from the N-terminus to the start of the HMM hit position. Proteins with mature domains <30 or >3000 amino acids were excluded.

S6.2 Control group selection

Control proteins were selected from the *B. bigelowii* proteome:

1. Exclude all proteins with HMM hits (uTP candidates)
2. Select candidates with lengths matching the mature domain length distribution
3. Submit to CELLO 2.5 (<http://cello.life.nctu.edu.tw/>) for subcellular localization prediction
4. Organism type: Eukaryote
5. Retain only proteins predicted as cytoplasmic or nuclear

This yielded 773 control proteins.

S6.3 Feature extraction

Features were extracted using ProtT5-XL-UniRef50 as described above, yielding 1024-dimensional embeddings for each mature domain.

S6.4 Classifier training

Parameter	Value
Algorithm	Logistic Regression
Regularization	L2 (C=1.0)
Class weights	Balanced
Cross-validation	5-fold stratified
Scaling	StandardScaler (mean=0, std=1)

S6.5 Statistical validation

Permutation test: Labels were randomly shuffled 1000 times. For each permutation, a classifier was trained and evaluated using 5-fold CV. The p-value is the fraction of permuted accuracies \geq observed accuracy.

Bootstrap confidence intervals: 1000 bootstrap samples were drawn with replacement. For each sample, the classifier was trained and evaluated, yielding a distribution of accuracy estimates. 95% CI computed as the 2.5th and 97.5th percentiles.

Full proteome validation: The classifier trained on the balanced dataset was applied to predict uTP status for all 44,363 *B. bigelowii* proteins. ROC AUC was computed using true labels (933 uTP, 43,430 non-uTP).

S7 Biophysical Property Analysis

S7.1 Property calculation

Properties were computed using BioPython’s ProteinAnalysis class:

Property	Method	Description
Isoelectric point	isoelectric_point()	pH at which net charge = 0
Instability index	instability_index()	Guruprasad et al. formula
GRAVY	gravy()	Grand average of hydropathy
Molecular weight	molecular_weight()	Sum of residue masses

S7.2 Disorder prediction

Intrinsic disorder was estimated using secondary structure propensity. Fraction coil was computed as the proportion of residues with low helix and sheet propensity based on the Chou-Fasman parameters.

S7.3 Effect size calculation

Cohen’s d was computed as:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{\text{pooled}}}$$

where $s_{\text{pooled}} = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}}$

Interpretation: $|d| = 0.2$ (small), $|d| = 0.5$ (medium), $|d| = 0.8$ (large).

S7.4 Multiple testing correction

Bonferroni correction was applied for 8 pre-specified property comparisons ($\alpha = 0.05/8 = 0.00625$).

S8 Gene Family Analysis

S8.1 Sequence representation

Each protein was represented as a k-mer frequency vector:

- k-mer size: 3
- Vector dimension: $20^3 = 8000$
- Normalization: L1 (frequencies sum to 1)

S8.2 Distance calculation

Pairwise Jaccard distance was computed:

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

where A and B are the sets of k-mers present in each sequence (frequency > 0).

S8.3 Hierarchical clustering

Clustering was performed using `scipy.cluster.hierarchy` with:

- Method: average linkage
- Distance threshold: 0.7 (corresponding approximately to 40% sequence identity)

S8.4 Permutation test

To test whether uTP proteins cluster more than expected by chance:

1. Compute observed metrics:
 - Fraction of uTP proteins sharing a family with another uTP protein
 - Number of distinct families containing uTP proteins
 - Maximum uTP proteins in any single family
2. Randomly reassign uTP labels among all proteins (preserving total count)
3. Recompute metrics
4. Repeat 10,000 times
5. P-value = fraction of permuted values \geq observed (for clustering metrics) or \leq observed (for diversity metrics)

S9 Functional Enrichment and Within-Category Analysis

S9.1 Functional annotation

Proteins were annotated using the existing *B. bigelowii* transcriptome annotations from Coale et al., which include COG (Clusters of Orthologous Groups) categories assigned via eggNOG-mapper.

S9.2 Within-category comparison

For each COG category with ≥ 10 uTP proteins and ≥ 10 control proteins:

1. Compute Cohen's d for each biophysical property within the category
2. Compare to overall effect size (pooling across categories)
3. Compute percent of effect explained by function:

$$\% \text{ explained} = \frac{d_{\text{overall}} - d_{\text{within}}}{d_{\text{overall}}} \times 100$$

Eight categories met the sample size criterion.

S9.3 Variance partitioning

Variance partitioning was performed using Type II ANOVA:

$$\text{Property} \sim \text{uTP_status} + \text{COG_category}$$

Variance components:

- **uTP unique:** $\text{SS}_{\text{uTP}}/\text{SS}_{\text{total}}$
- **Function unique:** $\text{SS}_{\text{COG}}/\text{SS}_{\text{total}}$
- **Shared:** Computed via sequential decomposition
- **Unexplained:** $\text{SS}_{\text{residual}}/\text{SS}_{\text{total}}$

S9.4 Meta-analysis of within-category effects

Heterogeneity across categories was assessed using the I^2 statistic:

$$I^2 = \frac{Q - (k - 1)}{Q} \times 100\%$$

where Q is Cochran's Q statistic and k is the number of categories.

Interpretation: $I^2 < 25\%$ (low), $25\% \leq I^2 < 75\%$ (moderate), $I^2 \geq 75\%$ (high heterogeneity).

S10 Code Availability

All analysis scripts are available at [repository URL]. The repository includes:

- `experiments/utp_motif_analysis/` – Motif discovery and coverage
- `experiments/utp_consensus_structure/` – Structure prediction and alignment
- `experiments/utp_sequence_clustering/` – Sequence space analysis
- `experiments/utp_structure_vs_null/` – Null model comparison
- `experiments/utp_presence_classifier/` – Mature domain classifier
- `experiments/utp_family_clustering/` – Gene family analysis
- `experiments/utp_functional_annotation/` – Functional enrichment

Each experiment directory contains a README.md with execution instructions and expected outputs.