SECISDesign - A Method to Design New and Recombinant Selenoproteins

Anke Busch and Rolf Backofen

Albert-Ludwigs-University Freiburg

Institute of Computer Science

Chair of Bioinformatics

Outline

- Biological Introduction
- The Computational Problem
- The Algorithm (SECISDesign)
- Results and Summary

Biological Introduction - 1

Selenoproteins:

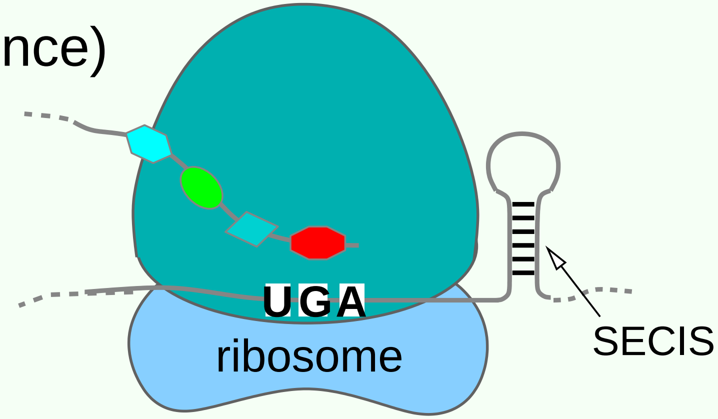
- contain selenium, incorporated as *selenocysteine* at the active site
- greatly enhanced enzymatic activities compared to the cysteine homologues

- important to human health:
  - thyroid hormone metabolism
  - immune function
  - protection against cancer

Biological Introduction - 2

Selenocysteine: (= 21*st* amino acid)

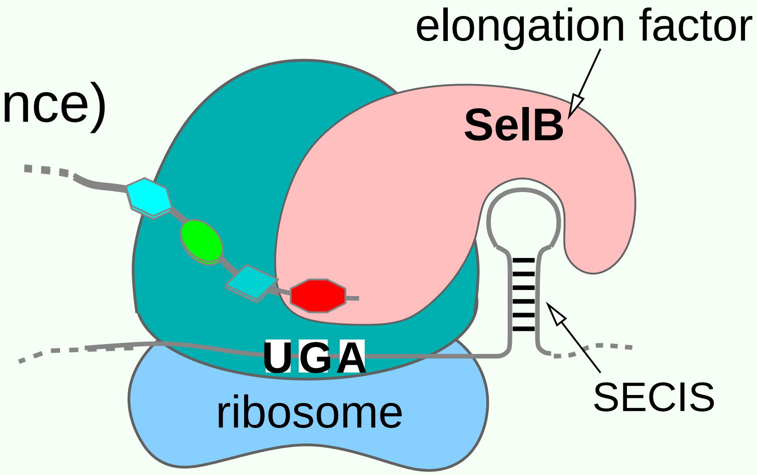
- encoded by the STOP-codon UGA
- inserted, if:
  - UGA is followed by a *SECIS-element* (hairpin-like structure + specific sequence)



Biological Introduction - 2

Selenocysteine: (= 21*st* amino acid)

- encoded by the STOP-codon UGA
- inserted, if:
    - UGA is followed by a *SECIS-element* (hairpin-like structure + specific sequence)
    - special elongation factor SelB available

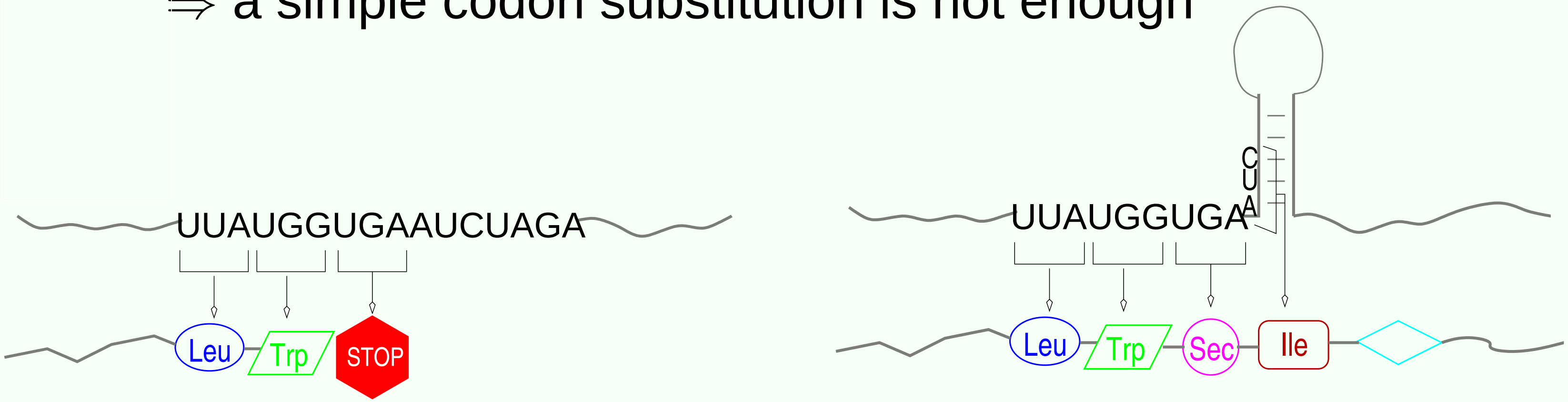


The Biological Problem - 1

Tasks:

1. Replacement of an amino acid (e.g. cysteine) by selenocysteine

⇒ a simple codon substitution is not enough



2. Expression of eukaryotic selenoproteins in *E.coli*

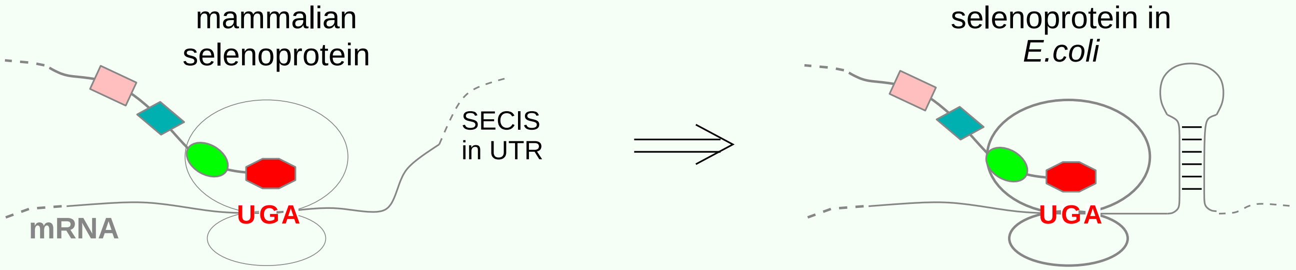
The Biological Problem - 2

Protein expression system: *E.coli*

Eukaryotes: SECIS-element *outside* the coding sequence

E.coli (bacteria): SECIS-element immediately downstream the UGA-Codon

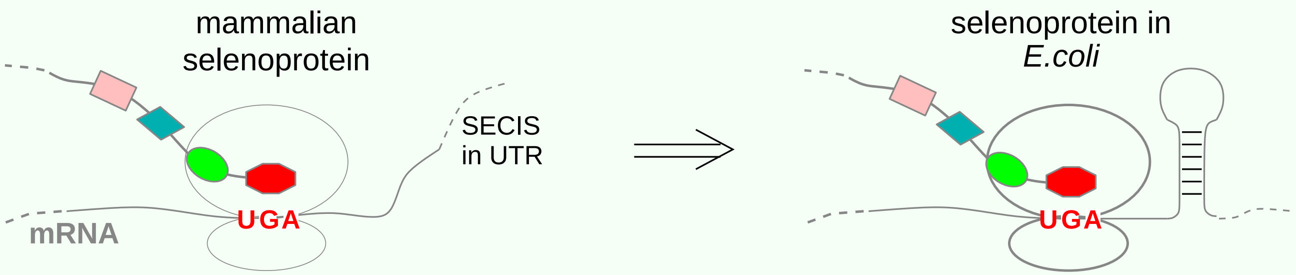
→ located *inside* the coding region of the protein



The Biological Problem - 2

Protein expression system: *E.coli*

Eukaryotes:	SECIS-element *outside* the coding sequence
E.coli (bacteria):	SECIS-element immediately downstream the UGA-Codon
	→ located *inside* the coding region of the protein

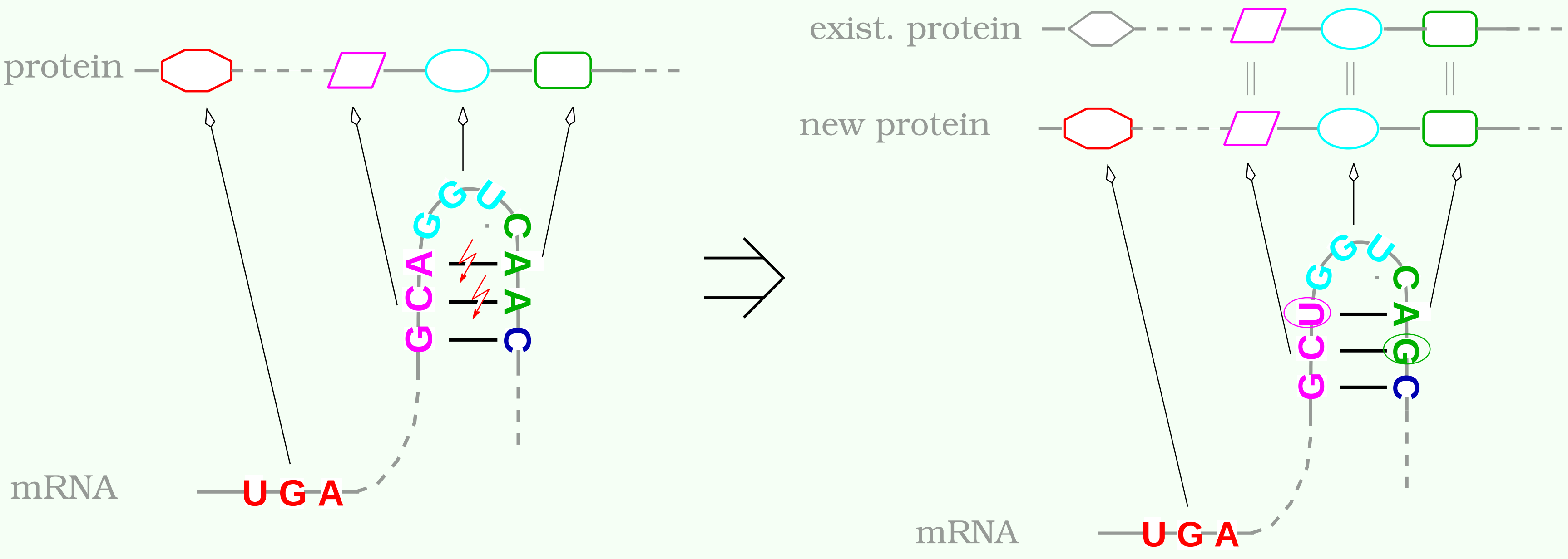


Consequence:

- eukaryotic selenoproteins can't be expressed directly in *E.coli*
- design of a SECIS-element next to the UGA-position
- this design may change the sequence of the protein

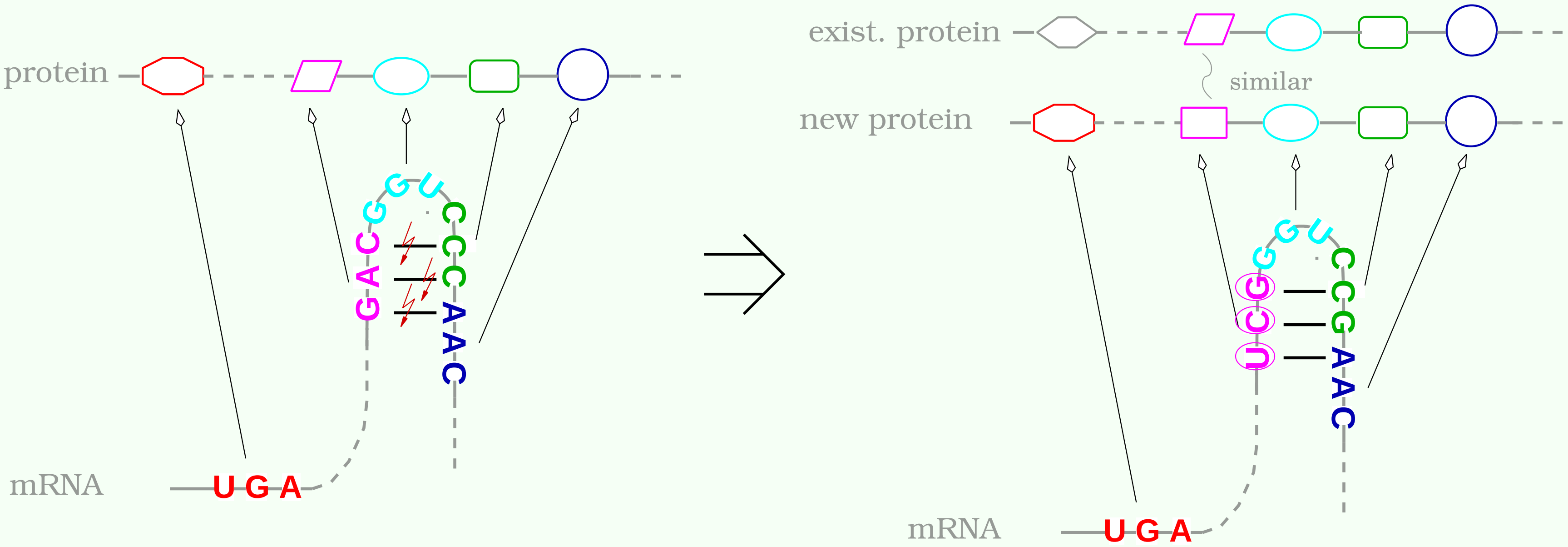
The Biological Problem - 3

SECIS-element-design: (sometimes without a change in the amino acid sequence...)



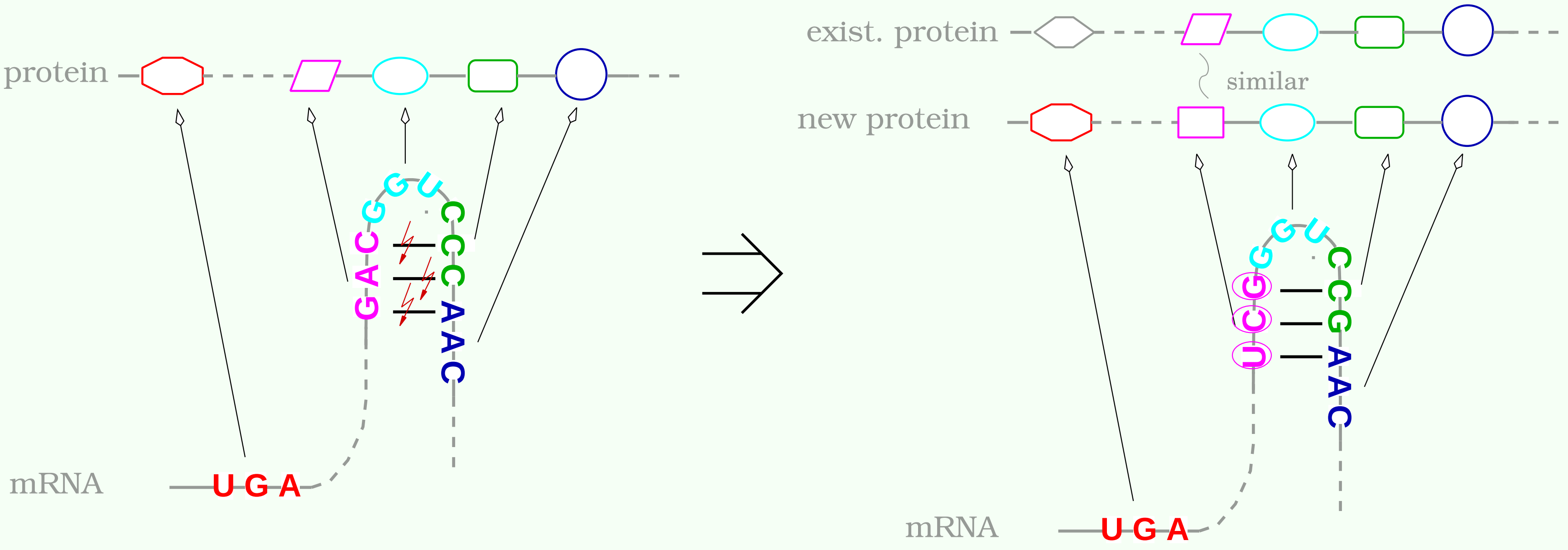
The Biological Problem - 4

...and sometimes wobble mutations are NOT enough!



The Biological Problem - 4

...and sometimes wobble mutations are NOT enough!

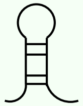


Problem:

compromise between

- quality of the SECIS-element and
- changes in the protein sequence

The Computational Problem

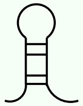
- **Given:** $G =$  ... typical SECIS secondary structure

  $S = S_1 \ldots S_{3n}$ ... nucleotide sequence (SECIS-consensus)

  $A = A_1 \ldots A_n$ ... original amino acid sequence

$$\begin{array}{cccccccc} A & = & A_1 & \ldots & A_i & \ldots & A_n \\ & & & & & & \\ S & = & S_1S_2S_3 & \ldots & S_{3i-2}S_{3i-1}S_{3i} & \ldots & S_{3n-2}S_{3n-1}S_{3n} \end{array}$$

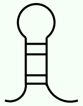
The Computational Problem

- **Given:** $G =$  ... typical SECIS secondary structure

  $S = S_1 \ldots S_{3n}$ ... nucleotide sequence (SECIS-consensus)

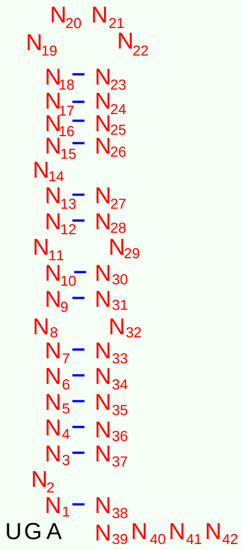
  $A = A_1 \ldots A_n$ ... original amino acid sequence

- **Wanted:** $N = N_1 \ldots N_{3n}$ ... mRNA sequence that

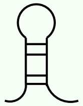
  · can adopt $G =$ 

$$A \;=\; A_1 \quad \ldots \quad A_i \quad \ldots \quad A_n$$

$$S \;=\; S_1 S_2 S_3 \quad \ldots \quad S_{3i-2} S_{3i-1} S_{3i} \quad \ldots \quad S_{3n-2} S_{3n-1} S_{3n}$$



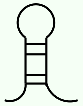
The Computational Problem

- **Given:** $G =$  ... typical SECIS secondary structure

  $S = S_1 \dots S_{3n}$ ... nucleotide sequence (SECIS-consensus)

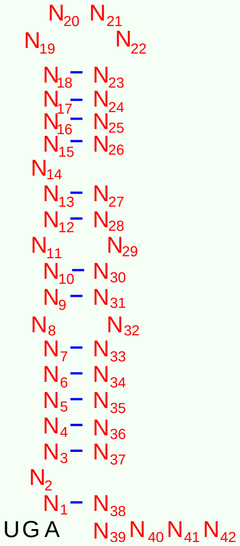
  $A = A_1 \dots A_n$ ... original amino acid sequence

- **Wanted:** $N = N_1 \dots N_{3n}$ ... mRNA sequence that

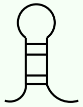
  · can adopt $G =$ 

  · has maximum similarity with S

$$\begin{array}{ccccccc}
A & = & A_1 & \dots & A_i & \dots & A_n \\
 & & & & & & \\
N & = & \overbrace{N_1N_2N_3} & \dots & \overbrace{N_{3i-2}N_{3i-1}N_{3i}} & \dots & \overbrace{N_{3n-2}N_{3n-1}N_{3n}} \\
 & & \sim\ \sim\ \sim & & \sim \quad \sim \quad \sim & & \sim \quad \sim \quad \sim \\
S & = & S_1S_2S_3 & \dots & S_{3i-2}S_{3i-1}S_{3i} & \dots & S_{3n-2}S_{3n-1}S_{3n}
\end{array}$$



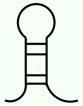
The Computational Problem

- **Given:** $G =$  ... typical SECIS secondary structure

  $S = S_1 \ldots S_{3n}$ ... nucleotide sequence (SECIS-consensus)

  $A = A_1 \ldots A_n$ ... original amino acid sequence

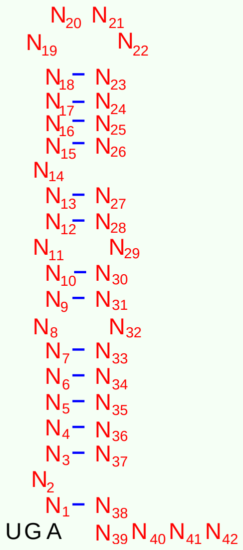
- **Wanted:** $N = N_1 \ldots N_{3n}$ ... mRNA sequence that

  · can adopt $G =$ 

  · has maximum similarity with S

  · encodes amino acid sequence A' with maximal similarity to A

$$
\begin{array}{ccccccc}
A & = & A_1 & \ldots & A_i & \ldots & A_n \\
 & & \sim & & \sim & & \sim \\
A' & = & A'_1 & \ldots & A'_i & \ldots & A'_n \\
N & = & \overbrace{N_1 N_2 N_3} & \ldots & \overbrace{N_{3i-2} N_{3i-1} N_{3i}} & \ldots & \overbrace{N_{3n-2} N_{3n-1} N_{3n}} \\
 & & \sim\ \sim\ \sim & & \sim\ \ \sim\ \ \sim & & \sim\ \ \sim\ \ \sim \\
S & = & S_1 S_2 S_3 & \ldots & S_{3i-2} S_{3i-1} S_{3i} & \ldots & S_{3n-2} S_{3n-1} S_{3n}
\end{array}
$$



Contrary Conditions

Possible Problem: contrary conditions at nucleotide and amino acid level

$$\begin{array}{c} A_i \\ \downarrow \\ Met \\ \| \\ \overbrace{A \quad U \quad G} \end{array}$$

→ essential amino acid

Contrary Conditions

Possible Problem: contrary conditions at nucleotide and amino acid level

$$\boxed{\begin{array}{c} A_i \\ \downarrow \\ Met \\ \shortparallel \\ \overbrace{A \quad U \quad G} \\ \neq \quad \neq \quad \neq \\ G \quad G \quad U \\ \uparrow \quad \uparrow \quad \uparrow \\ S_{3i-2}S_{3i-1}S_{3i} \end{array}}$$

→ essential amino acid

→ essential SECIS-sequence

$\Rightarrow$ insertions/deletions & *optional* bonds

Extentions - 1

(1) Insertions and deletions (amino acids)

- single amino acids are allowed to be deleted or inserted
- mapping between A and A' changes

$$\begin{array}{llll}
A: & A_1 & - & A_2 \\
A': & A'_1 & A'_2 & A'_3 \\
N: & \overbrace{N_1N_2N_3} & \overbrace{N_4N_5N_6} & \overbrace{N_7N_8N_9} \\
S: & \underbrace{S_1S_2S_3} & \underbrace{S_4S_5S_6} & \underbrace{S_7S_8S_9} \\
\text{SECpos} & 1 & 2 & 3
\end{array}$$

- insertion of A'_2
- A'_2 has no counterpart in A
- insertion penalty IP

Extentions - 1

(1) Insertions and deletions (amino acids)

- single amino acids are allowed to be deleted or inserted
- mapping between A and A' changes

$$
\begin{array}{lccc}
A: & A_1 & - & A_2 \\
A': & A'_1 & A'_2 & A'_3 \\
N: & \overbrace{N_1N_2N_3} & \overbrace{N_4N_5N_6} & \overbrace{N_7N_8N_9} \\
S: & \underbrace{S_1S_2S_3} & \underbrace{S_4S_5S_6} & \underbrace{S_7S_8S_9} \\
\text{SECpos} & 1 & 2 & 3
\end{array}
$$

$$
\begin{array}{lcccc}
A: & A_1 & A_2 & A_3 & A_4 \\
A': & A'_1 & - & A'_2 & A'_3 \\
N: & \overbrace{N_1N_2N_3} & - - - & \overbrace{N_4N_5N_6} & \overbrace{N_7N_8N_9} \\
S: & \underbrace{S_1S_2S_3} & \multicolumn{2}{c}{\underbrace{- - - \quad S_4S_5S_6}} & \underbrace{S_7S_8S_9} \\
\text{SECpos} & 1 & \multicolumn{2}{c}{2} & 3
\end{array}
$$

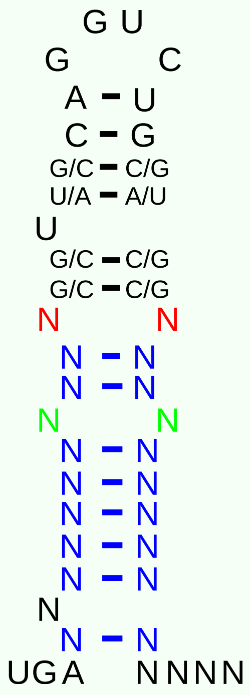
- insertion of A'_2
- A'_2 has no counterpart in A
- insertion penalty IP

- deletion of A_2
- compare $N_4N_5N_6$ with A_3
- deletion penalty DP

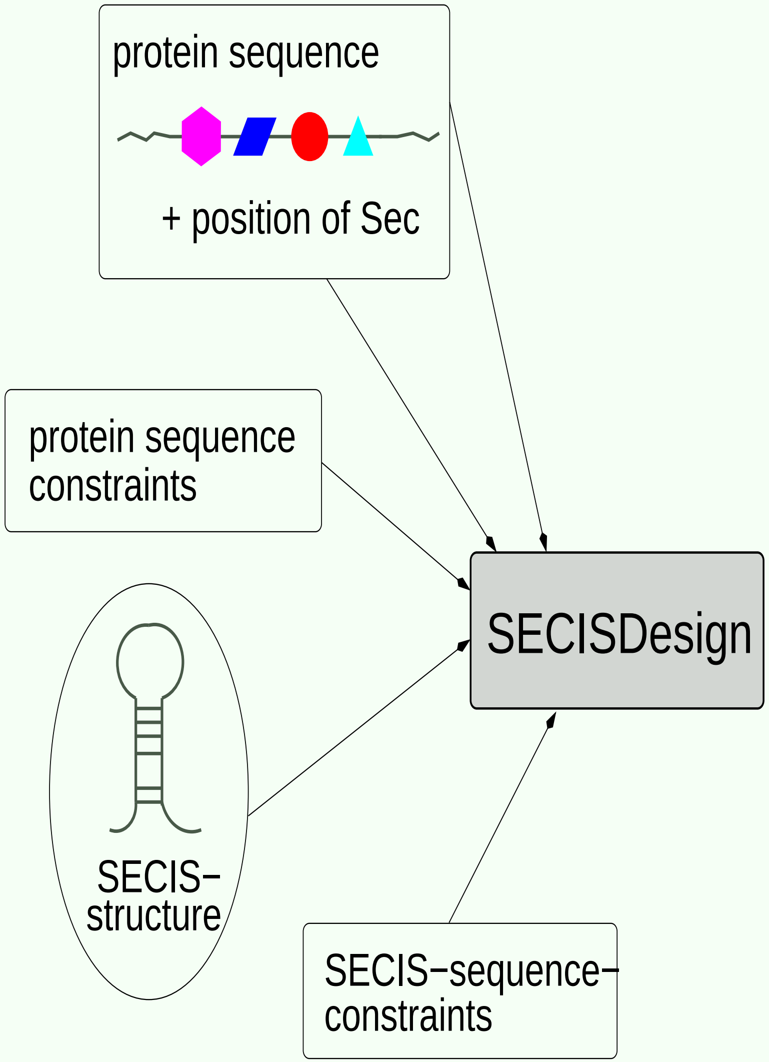
Extentions - 2

(2) Kinds of bonds in the SECIS-Element (mRNA)

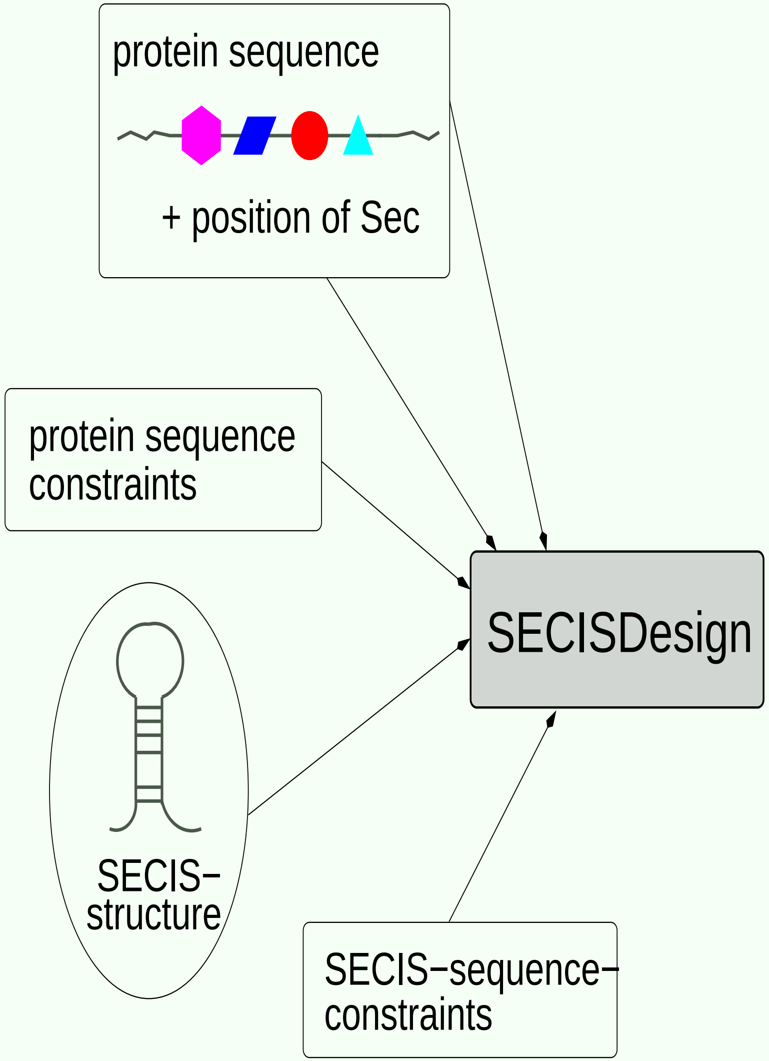
- *bond* ... mandatory bond
- *optional bond* ... not necessary but of advantage, if formed
- *prohibited bond* ... a bond that is not allowed
- *unfavorable bond* ... not necessary but of advantage, if NOT formed



Input of SECISDesign



Input of SECISDesign



- **similarity functions** $f_i(L_i, a_i, t_i)$,

where L_i ... *codon* corresponding to $N_{3i-2}N_{3i-1}N_{3i}$

$t_i \in \{-1$ (deletion)$, 0$ (subst.)$, +1$ (insertion)$\}$

$a_i = \sum_{j=1}^{i} t_j$

including: · *similarity at nucleotide level*

· *similarity at amino acid level*

The Algorithm - Overview

Two Steps:

1. *Dynamic Programming:* (a) divides the problem in subproblems

   (b) solves subproblems and stores results

   (c) combines the sub-solutions

- finds an *optimal* RNA sequence conc. *similarity* (nucleotide + amino acid) that can fold into the *SECIS-structure*

  (maximizes $\sum_{i=1}^{n} f_i(L_i, a_i, t_i)$)

The Algorithm - Overview

Two Steps:

1. *Dynamic Programming:* (a) divides the problem in subproblems
   (b) solves subproblems and stores results
   (c) combines the sub-solutions
   - finds an *optimal* RNA sequence conc. *similarity* (nucleotide + amino acid) that can fold into the *SECIS-structure* (maximizes $\sum_{i=1}^{n} f_i(L_i, a_i, t_i)$)

2. *Local Search:*
   - further improve the designed sequence to *increase the folding probability (inverse RNA folding)*

Dynamic Programming - 1

Central Function:

$$\left[\begin{array}{lll} L_i & \ldots & \text{assignment of the } i\text{-th codon} \\ l & \ldots & \text{\#insertions - \#deletions left} \\ s & \ldots & \text{\#insertions - \#deletions inside interval} \end{array}\right]$$

$$w_j^i(L_i, L_j, m, l, s)$$

$$= \max_{\substack{L_{i+1}\ldots L_{j-1} \\ t(s)}} \left\{ \sum_{i<k\leq j} f_k\Big(L_k, l + \sum_{g=1}^{k-i} t_g, t_{k-i}\Big) \;\middle|\; \begin{array}{l} L_i\ldots L_j \text{ satisfy SECIS-graph,} \\ m \text{ real. opt. bonds at } L_i\ldots L_j \end{array} \right\}$$

Dynamic Programming - 1

Central Function:

$$\left[\begin{array}{lll} L_i & \ldots & \text{assignment of the } i\text{-th codon} \\ l & \ldots & \text{\#insertions - \#deletions left} \\ s & \ldots & \text{\#insertions - \#deletions inside interval} \end{array}\right]$$

$$w_j^i(L_i, L_j, m, l, s) = \max_{\substack{L_{i+1}\ldots L_{j-1} \\ t(s)}} \left\{ \sum_{i<k\leq j} f_k\left(L_k, l + \sum_{g=1}^{k-i} t_g, t_{k-i}\right) \;\middle|\; \begin{array}{l} L_i \ldots L_j \text{ satisfy SECIS-graph,} \\ m \text{ real. opt. bonds at } L_i \ldots L_j \end{array} \right\}$$

Value of Interest:

$$\max_{\substack{L_1, L_n, m \\ |l| \leq 1, |s| \leq n-1}} \{w_n^1(L_1, L_n, m, l, s) + f_1(L_1, l, l)\}$$

$\Rightarrow$ solved by dynamic programming

Dynamic Programming - 2

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l + s_1, s_2) \right) & \text{otherwise} \end{cases}$$

where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

G U
G C
U
G

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

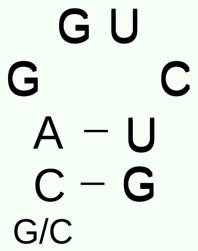
$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l + s_1, s_2) \right) & \text{otherwise} \end{cases}$$

where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l + s_1, s_2) \right) & \text{otherwise} \end{cases}$$

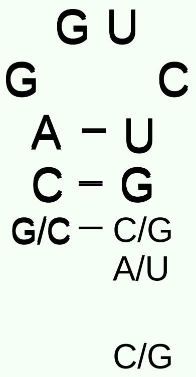


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i + 1$

Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \text{splits } (m_1, m_2) \text{ of } m \\ \text{splits } (s_1, s_2) \text{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l + s_1, s_2) \right) & \text{otherwise} \end{cases}$$

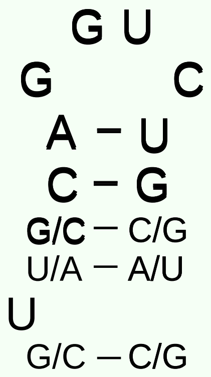


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i + 1$

Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l + s_1, s_2)\right) & \text{otherwise} \end{cases}$$

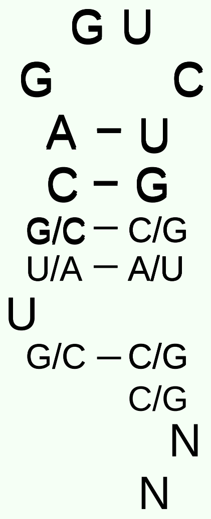


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i + 1$

Dynamic Programming - 2

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \text{splits } (m_1, m_2) \text{ of } m \\ \text{splits } (s_1, s_2) \text{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2) \right) & \text{otherwise} \end{cases}$$

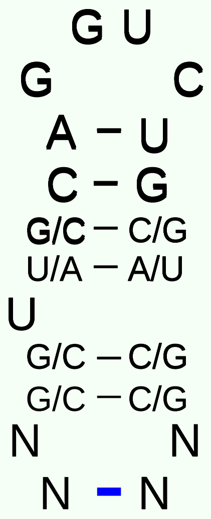


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2) \right) & \text{otherwise} \end{cases}$$

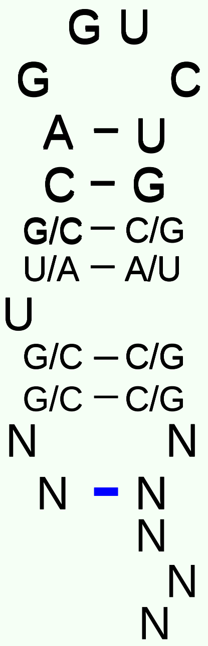

G U
G C
A – U
C – G
G/C – C/G
U/A – A/U
U
G/C – C/G
G/C – C/G
N N
N – N

where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w^{i}_{i+k}}(L_i, L_{i+k}, m, l, s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l + s_1, s_2) \right) & \text{otherwise} \end{cases}$$

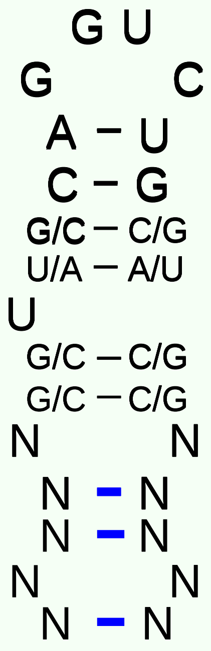


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\[2ex] \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2) \right) & \text{otherwise} \end{cases}$$

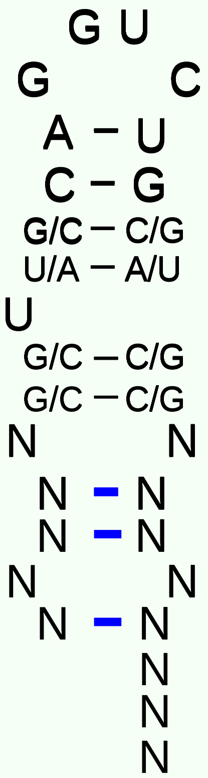


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1,m_2) \textit{ of } m \\ \textit{splits } (s_1,s_2) \textit{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2) \right) & \text{otherwise} \end{cases}$$

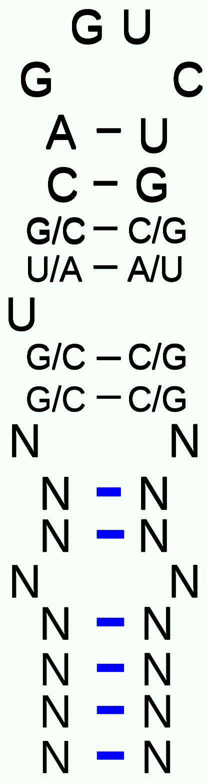


where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \textit{splits } (m_1, m_2) \textit{ of } m \\ \textit{splits } (s_1, s_2) \textit{ of } s}} \left(\mathbf{w}^i_p(L_i, L_p, m_1, l, s_1) + \mathbf{w}^p{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2)\right) & \text{otherwise} \end{cases}$$



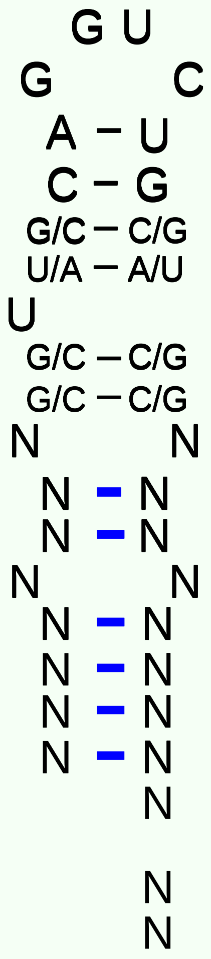
where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

Dynamic Programming - 2

Recurrence Theorem: $\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) =$

$$\begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \text{splits } (m_1,m_2) \text{ of } m \\ \text{splits } (s_1,s_2) \text{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2) \right) & \text{otherwise} \end{cases}$$

where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

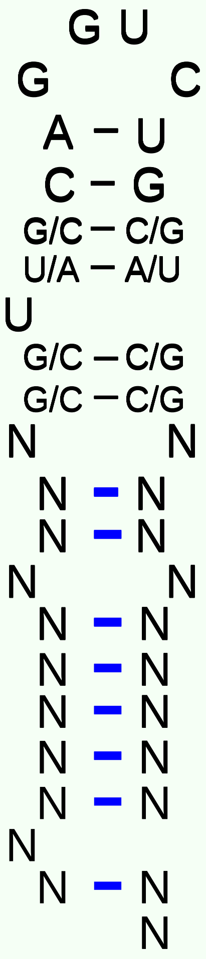


Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i,L_{i+k},m,l,s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \text{splits } (m_1,m_2) \text{ of } m \\ \text{splits } (s_1,s_2) \text{ of } s}} \left(\mathbf{w}^i_p(L_i,L_p,m_1,l,s_1)+\mathbf{w}^p{}_{i+k}(L_p,L_{i+k},m_2,l+s_1,s_2)\right) & \text{otherwise} \end{cases}$$

where $p \in \{i+1,...,j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$

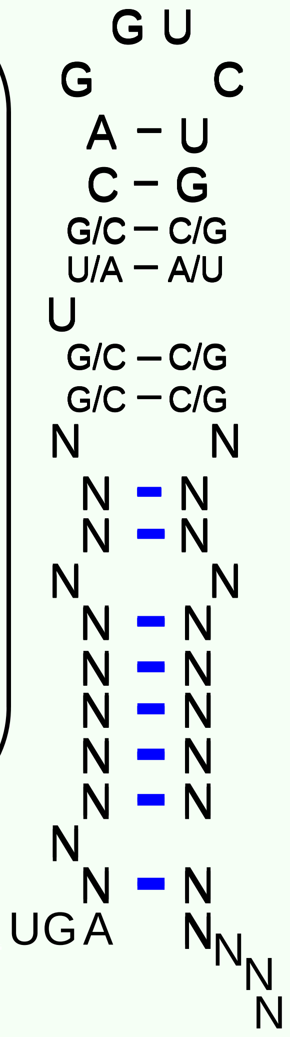


Dynamic Programming - 2

Recurrence Theorem:

$$\mathbf{w}^{\mathbf{i}}_{\mathbf{i+k}}(L_i, L_{i+k}, m, l, s) = \begin{cases} -\infty & \text{if } L_i \text{ and } L_{i+k} \text{ contradict the SECIS-constraints} \\ \max\limits_{\substack{L_p \\ \text{splits } (m_1, m_2) \text{ of } m \\ \text{splits } (s_1, s_2) \text{ of } s}} \left(\mathbf{w}^{i}_{p}(L_i, L_p, m_1, l, s_1) + \mathbf{w}^{p}{}_{i+k}(L_p, L_{i+k}, m_2, l+s_1, s_2) \right) & \text{otherwise} \end{cases}$$

where $p \in \{i+1, ..., j-1\}$ ... farthest codon having a bond with the i-th codon, if there is none, $p = i+1$



Local Search

After Step I:

Found a sequence that

- ***can*** fold into the SECIS-structure
- has ***maximal*** similarity to the SECIS-sequence
- encodes an amino acid sequence having ***maximal*** similarity to the original protein

Local Search

After Step I:

Found a sequence that

- ***can*** fold into the SECIS-structure
- has ***maximal*** similarity to the SECIS-sequence
- encodes an amino acid sequence having ***maximal*** similarity to the original protein

⇓

Step II: *(Local Search)*

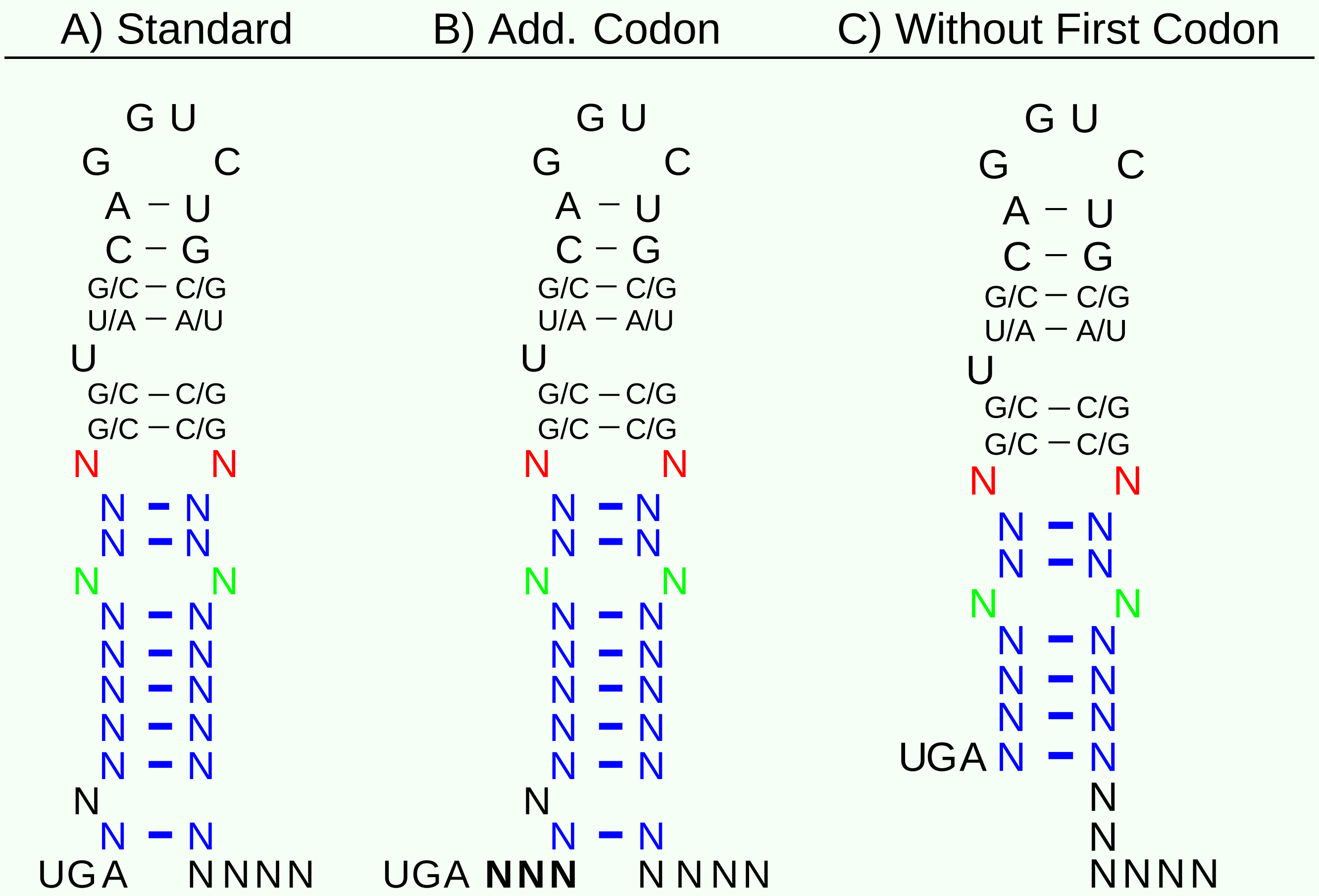
Local mutations to increase the folding probability (+ keep minimal similarities)

Methods:

- adaptive walk (as used in RNAinverse of the Vienna RNA Package)
- full local search
- stochastic local search

SECIS-elements

Alternatives of a SECIS-element in E.coli: *(FdhF)*



A) Standard

G U
G C
A – U
C – G
G/C – C/G
U/A – A/U
U
G/C – C/G
G/C – C/G
N N
N – N
N – N
N N
N – N
N – N
N – N
N – N
N – N
N
N – N
UGA NNNN

B) Add. Codon

G U
G C
A – U
C – G
G/C – C/G
U/A – A/U
U
G/C – C/G
G/C – C/G
N N
N – N
N – N
N N
N – N
N – N
N – N
N – N
N – N
N
N – N
UGA **NNN** NNNN

C) Without First Codon

G U
G C
A – U
C – G
G/C – C/G
U/A – A/U
U
G/C – C/G
G/C – C/G
N N
N – N
N – N
N N
N – N
N – N
N – N
UGA N – N
N
N
NNNN

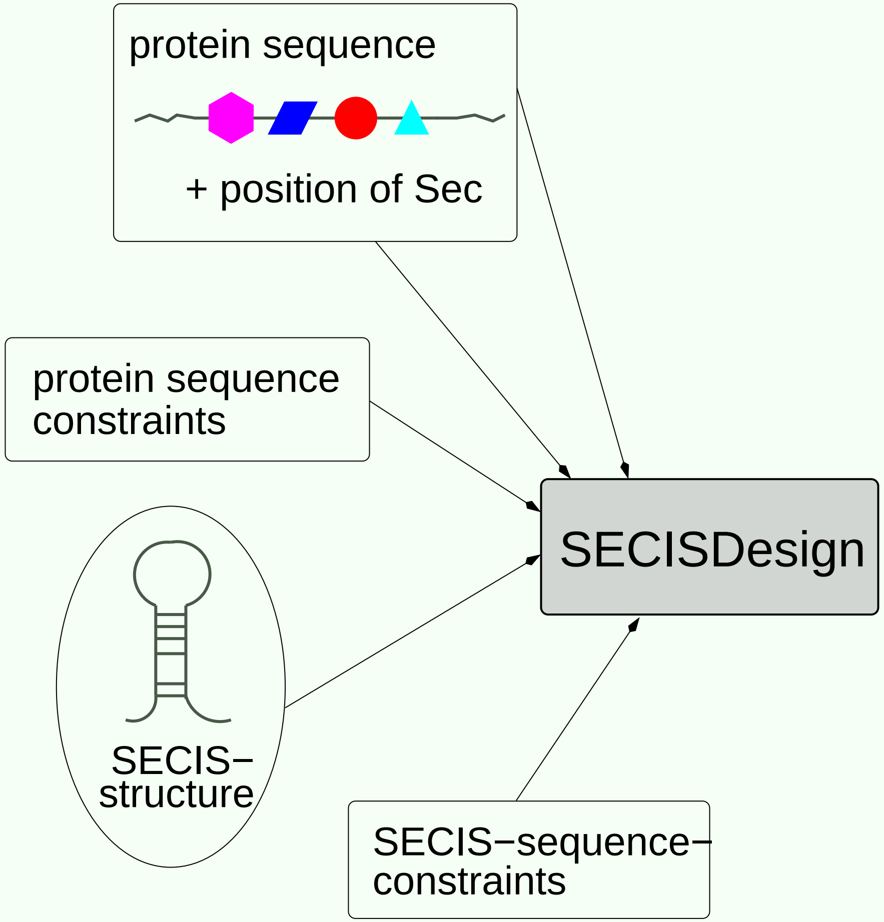
from Liu et al., NAR 1998

Results

Methionine Sulfoxide Reductase B (MsrB): (also analyzed in Bar-Noy et al., 2002)

	Amino Acid Sequences (starting after pos. of Sec)	*Sim. (BLOSUM62)*	*mRNA-Sequences and -Structures*	*Prob.*
mouse MsrB	**IFSSSLKFVPKGKE**			
MsrB_BaNo	**IFSTVAGLHPKGKE**	35	AUAUUUAGCACGGUUGCAGGUCUGCACCCUAAAGGCAAAGAA ....((.((((....)))))))..............	0.01
mouse MsrB	**IFSSSLKFVPKGKE**			
MsrB_1A	**IFSSLPGLVPKGKE**	43	AUUUUCUCUUCGCUACCAGGUCUGGUGCCAAAAGGAAAAGAA .((((((.((.((.((((....)))))).)).))))))....	0.19
MsrB_2A	**IFSSLPGLVPQGAE**	33	AUCUUCUCGUCGCUACCAGGUCUGGUGCCACAAGGAGCCGAA ..(((((.((.((.((((....)))))).)).))))).....	**0.75**
mouse MsrB	**IFSSSLK-FVPKGKE**			
MsrB_1B	**IFSSSLPGLVPKGKE**	57+*IP*	AUAUUUUCCUCUUCGCUACCAGGUCUGGUGCCAAAAGGAAAAGAA ....((((((.((.((.((((....)))))).)).))))))....	0.08
MsrB_2B	**IVSSSLPGLVPQGAE**	40+*IP*	AUAGUCUCCUCGUCGCUACCAGGUCUGGUGCCACAAGGAGCAGAA ...(.(((((.((.((.((((....)))))).)).))))))....	**0.62**
mouse MsrB	**IFSSSLKFVPKGKE**			
MsrB_1C	**IFSLP-GLVPKGKE**	41+*DP*	AUCUUUUCGCUACCAGGUCUGGUGCCAAAAGGUAAAGAA (((((((.((.((((....)))))).)))))))......	**0.64**

The Algorithm - Summary



protein sequence

+ position of Sec

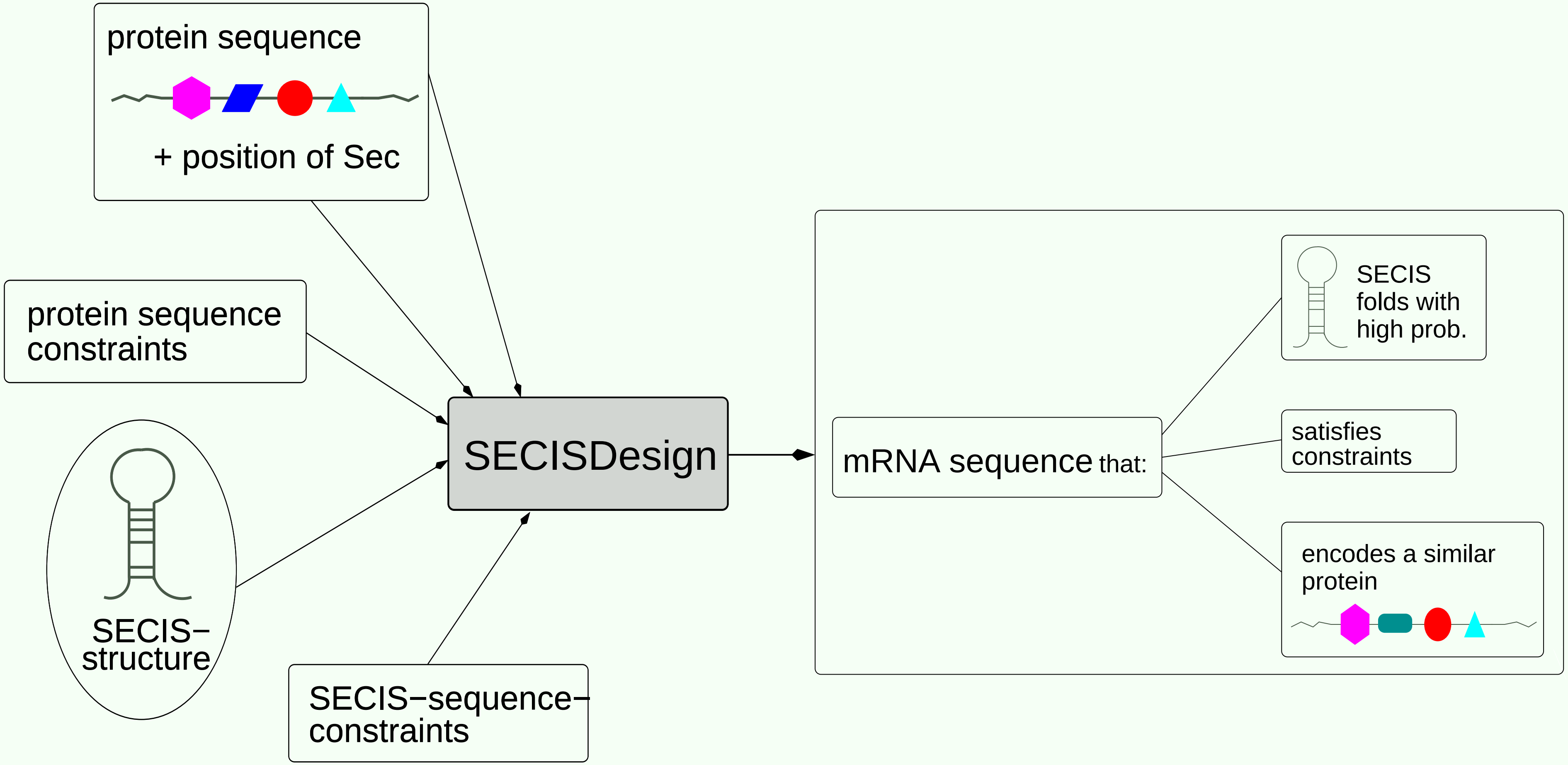
protein sequence constraints

SECIS−structure

SECIS−sequence−constraints

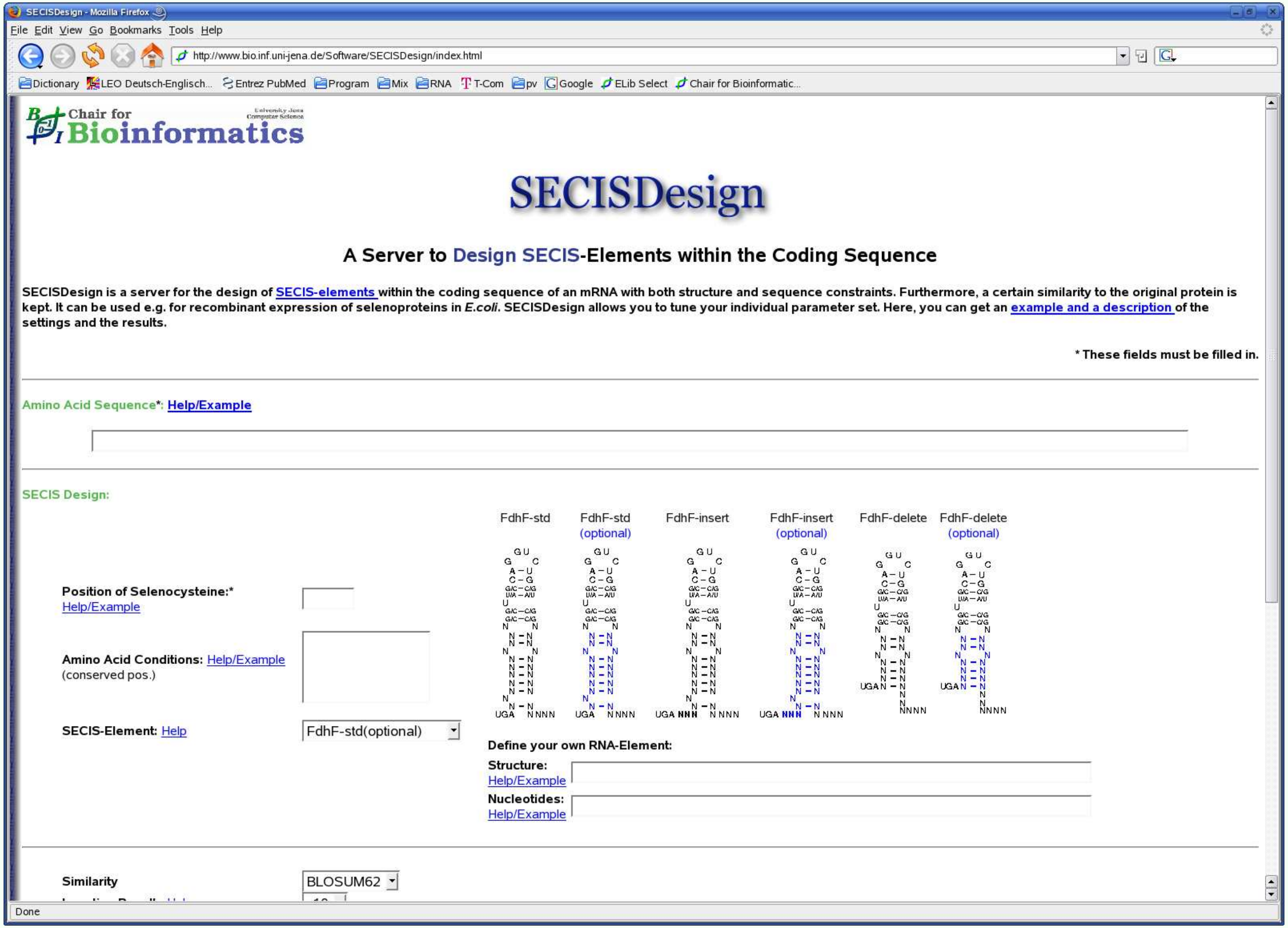
SECISDesign

The Algorithm - Summary


protein sequence
+ position of Sec
protein sequence constraints
SECIS−structure
SECIS−sequence−constraints
SECISDesign
mRNA sequence that:
SECIS folds with high prob.
satisfies constraints
encodes a similar protein

Webinterface

http://www.bioinf.uni-freiburg.de → Software → SECISDesign


SECISDesign - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.bio.inf.uni-jena.de/Software/SECISDesign/index.html
Dictionary
LEO Deutsch-Englisch...
Entrez PubMed
Program
Mix
RNA
T-Com
pv
Google
ELib Select
Chair for Bioinformatic...
Chair for Bioinformatics
University Jena Computer Science
Done

SECISDesign

A Server to Design SECIS-Elements within the Coding Sequence

SECISDesign is a server for the design of SECIS-elements within the coding sequence of an mRNA with both structure and sequence constraints. Furthermore, a certain similarity to the original protein is kept. It can be used e.g. for recombinant expression of selenoproteins in *E.coli*. SECISDesign allows you to tune your individual parameter set. Here, you can get an example and a description of the settings and the results.

* These fields must be filled in.

Amino Acid Sequence*: Help/Example

SECIS Design:

Position of Selenocysteine:* Help/Example

Amino Acid Conditions: Help/Example (conserved pos.)

SECIS-Element: Help FdhF-std(optional)

Define your own RNA-Element:

Structure: Help/Example

Nucleotides: Help/Example

Similarity BLOSUM62

Thank you for your attention.

Further reading:

Busch,A., S. Will, R. Backofen (2005). SECISDesign - A Server to Design SECIS-Elements within the Coding Sequence. *Bioinformatics*, **21(15)**, 3312-3.

Backofen, R. and Busch, A. (2004). Computational Design of New and Recombinant Selenoproteins. *Proc. of the 15th Annual Symposium on Combinatorial Pattern Matching (CPM2004)*.