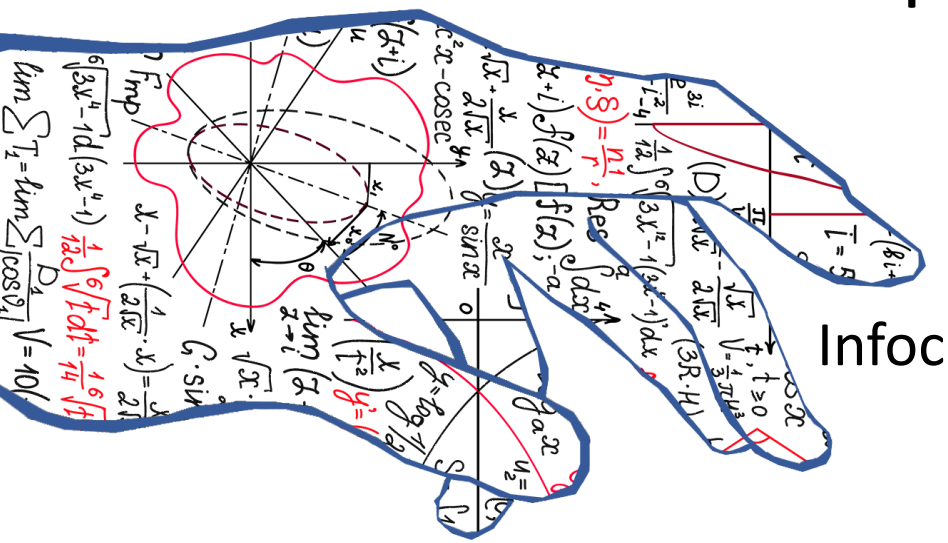
ITMO UNIVERSITY

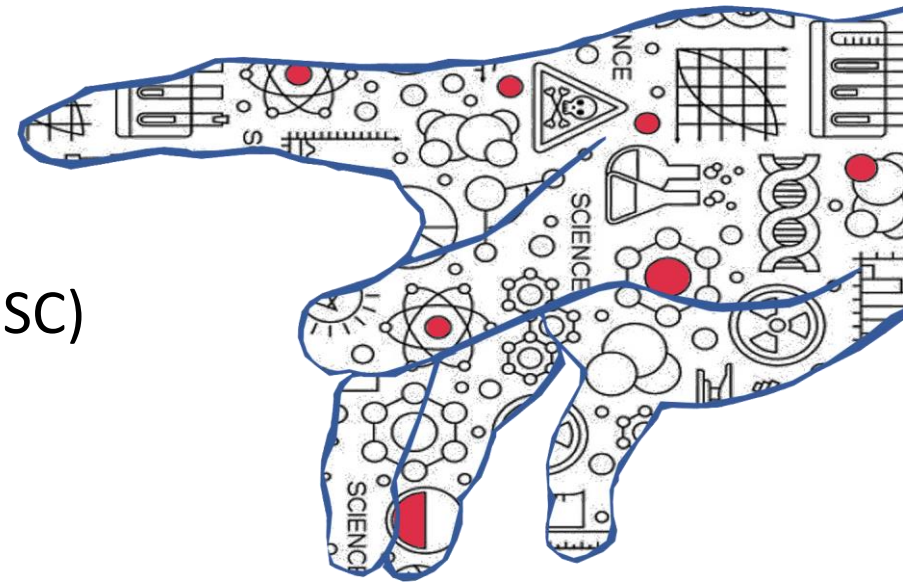
INFOCHEMISTRY SCIENTIFIC CENTER

Cheminformatics and synthetic biology: computational methods and projects

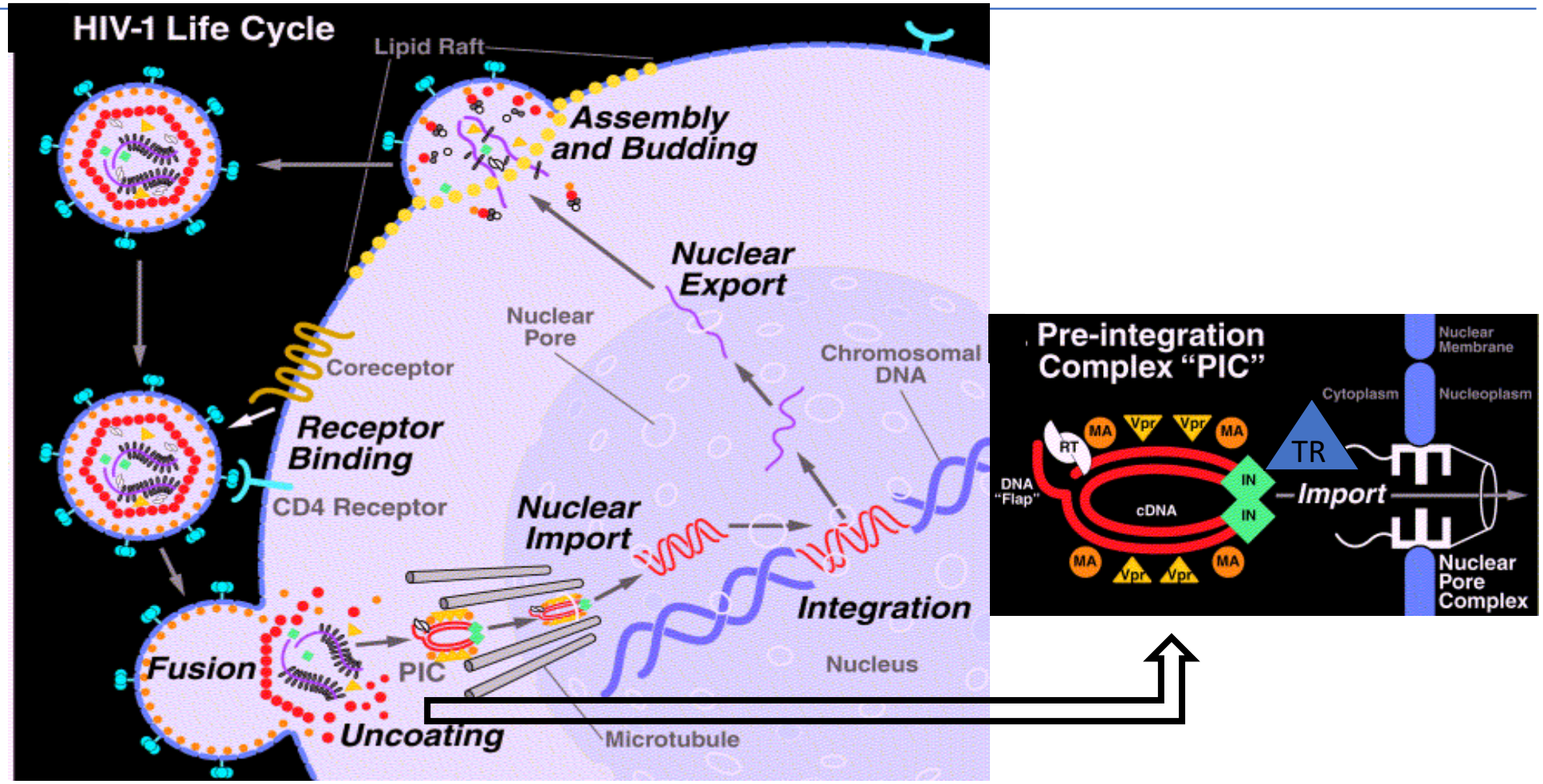
Prof. Sergey Shityakov



Infochemistry Scientific Center (ISC)
ITMO University
Saint-Petersburg, 2024



HIV-1 PIC



Sherman et al., Micro. Inf, 2002

Synthetic biology of HIV

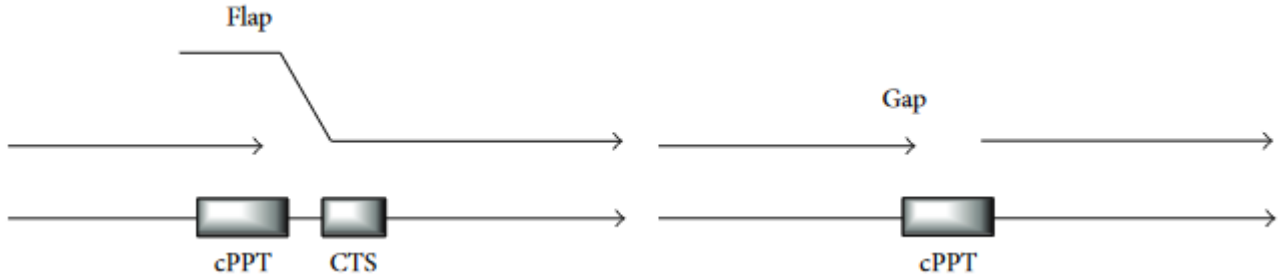


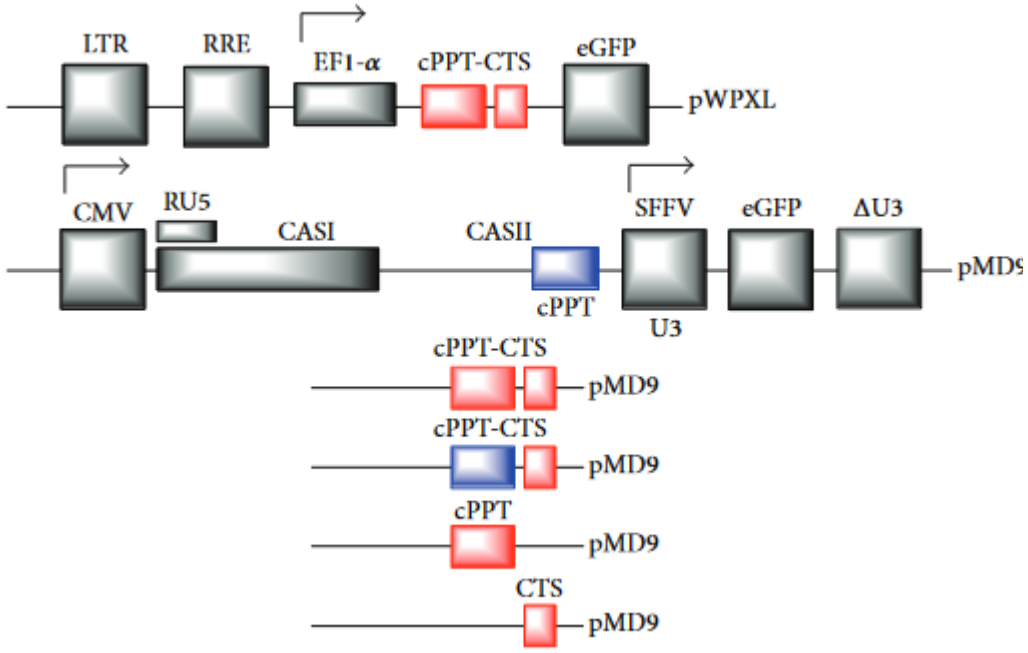
TABLE 1: Sequences of the viral structural elements used in the construction of modified expression vectors.

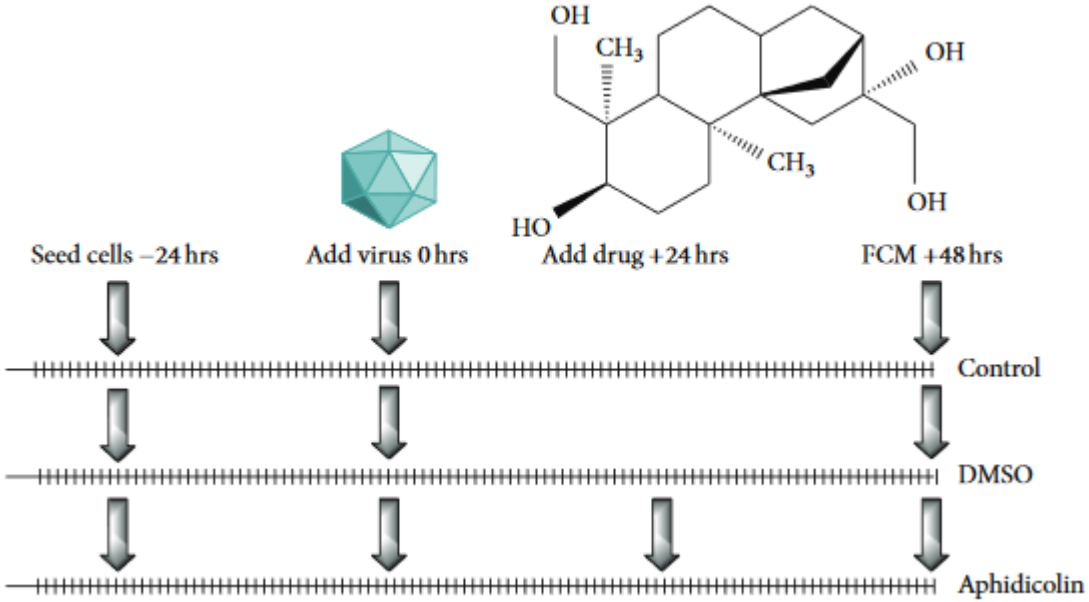
Element	Length (bp)	Sequence
cPPT (PFV)	9	AGGAGAGGG
cPPT (HIV)	33	ATCCACAATTTTAAAAGAAAAGGGGGGATTGGG
CTS	16	AAAAATTCAAAATTTT

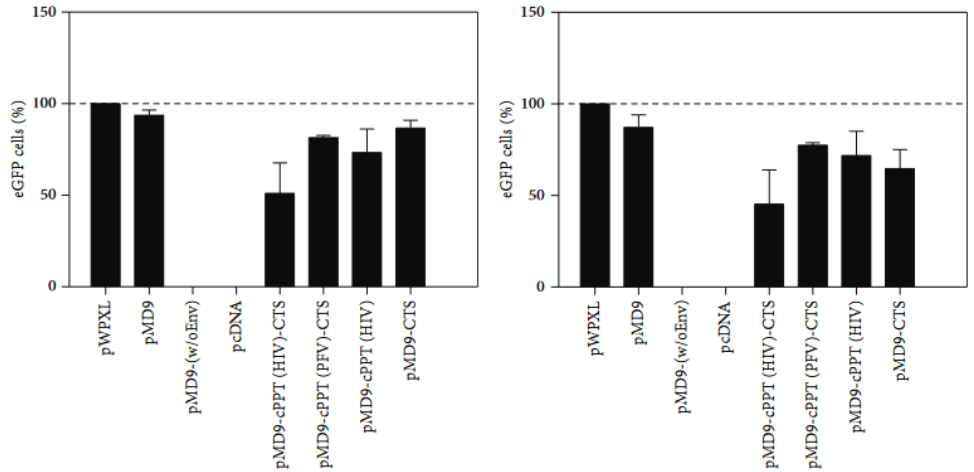
TABLE 2: Forward and reverse primers for a PCR amplification of the fragments used in the construction of modified expression vectors.

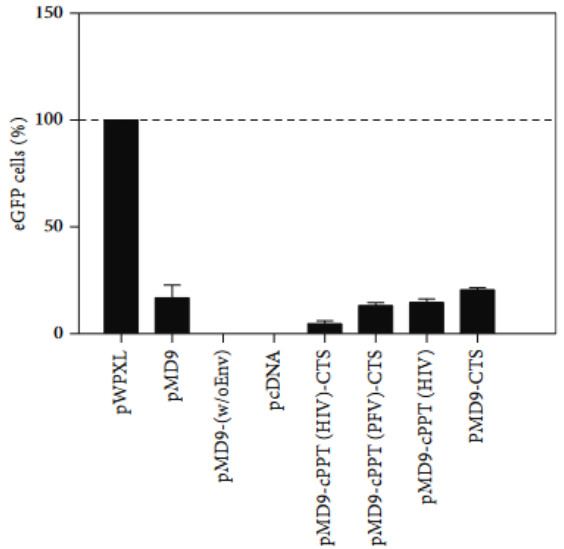
Fragments	Primers
cPPT (PFV)-CTS	5′-TATACAATTGCAGGAGAGGGATTGGGGGGTACAGTGCAG-3′
	5′-TATAAGGCCTCTGTCCCTGTAATAAACC-3′
cPPT (HIV)-CTS	5′-TATACAATTGATGGCAGTATCCAC-3′
	5′-TATAAGGCCTCTGTCCCTGTAATAAACC-3′
cPPT (HIV)	5′-TATACAATTGATGGCAGTATCCAC-3′
	5′-TATAAGGCCTGTAATTTGTTTTTGTAATTCT-3′
CTS	5′-TATATACAATTGGGGGGTACAGTGCAGGGG-3′
	5′-TATATAAGGCCTTCCCTGTAAACCCGAAAATTTTG-3′

Synthetic biology of HIV

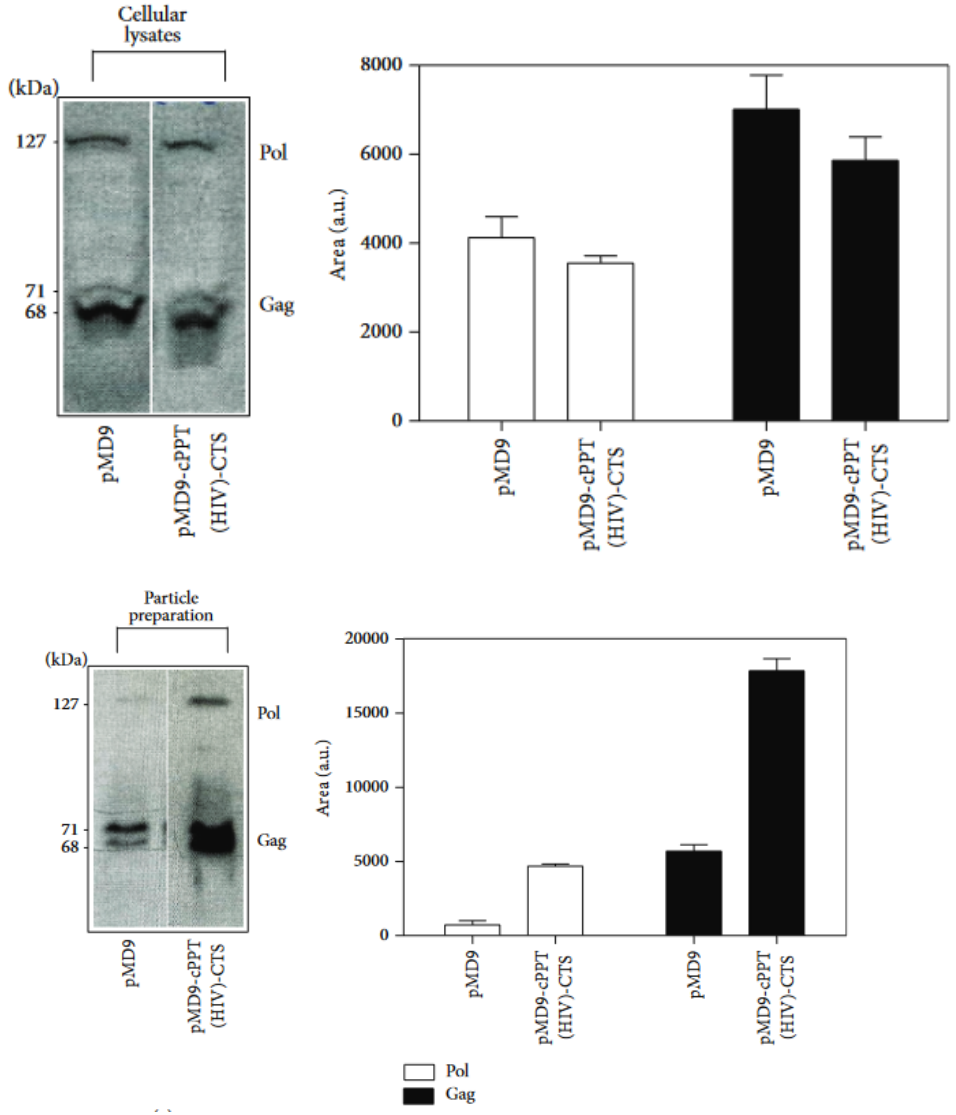


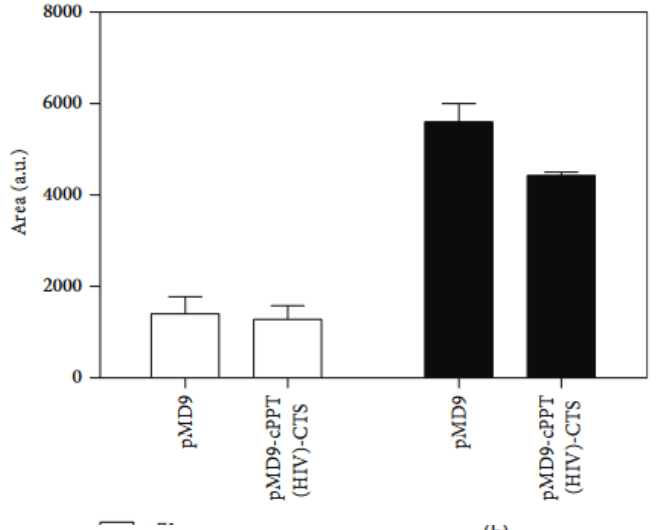




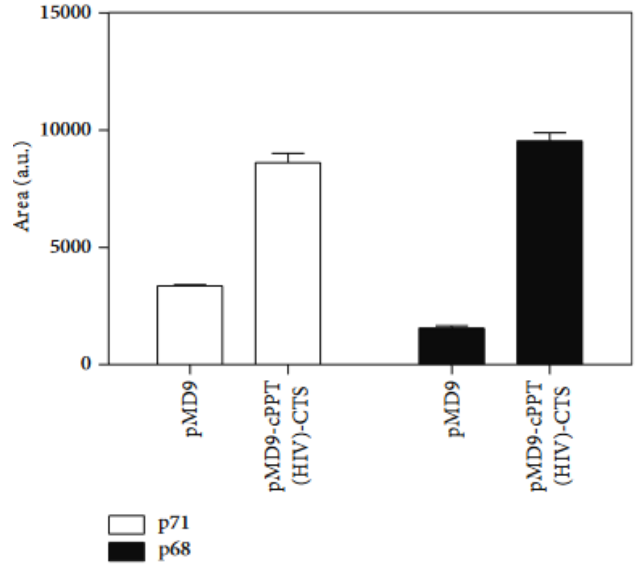


Synthetic biology of HIV

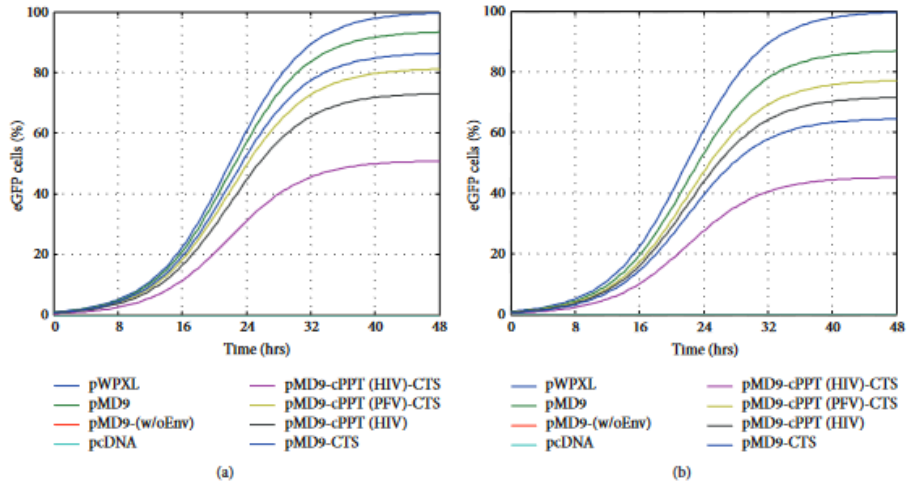


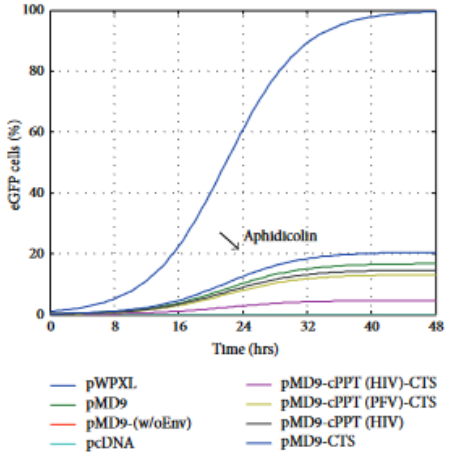


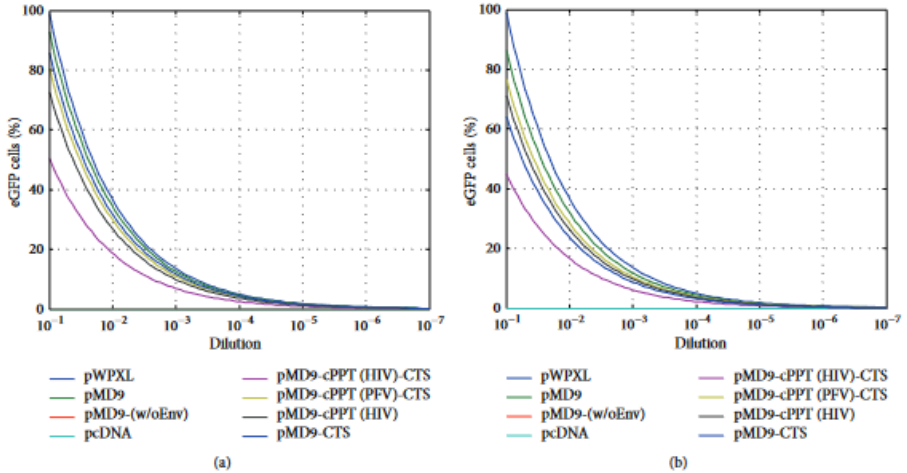
(b)

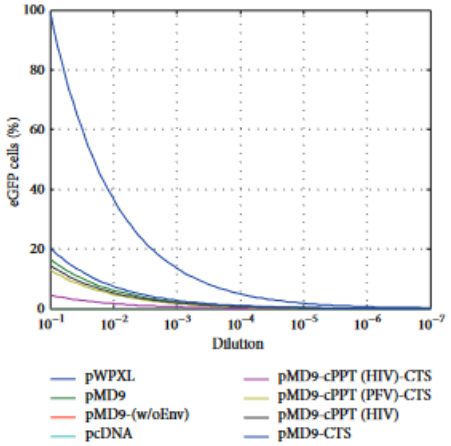


Synthetic biology of HIV





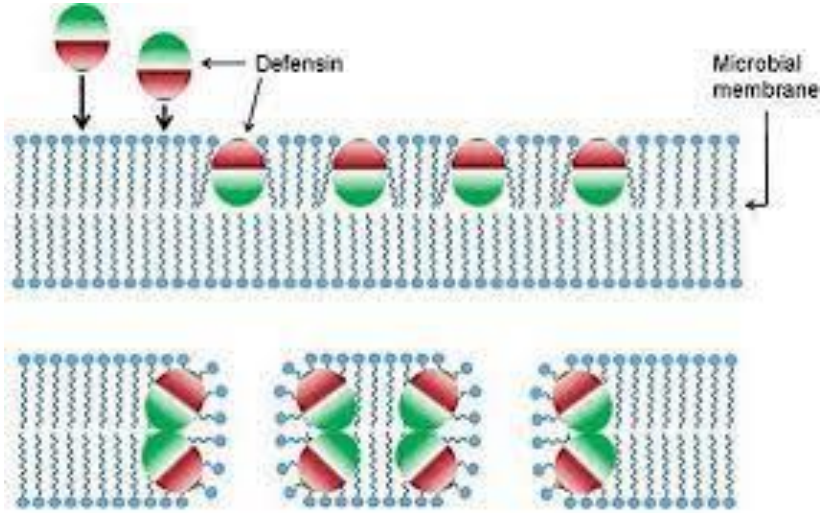


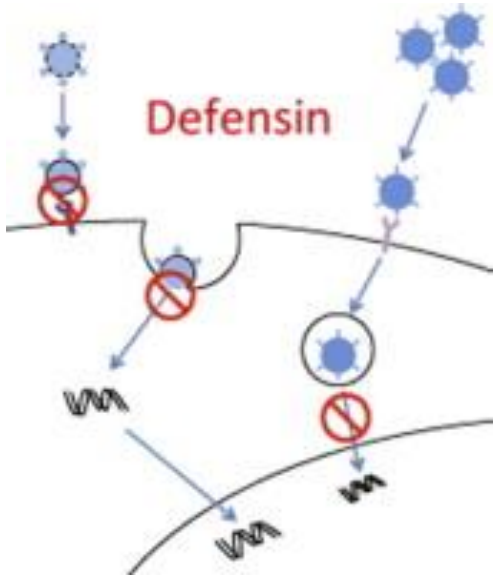


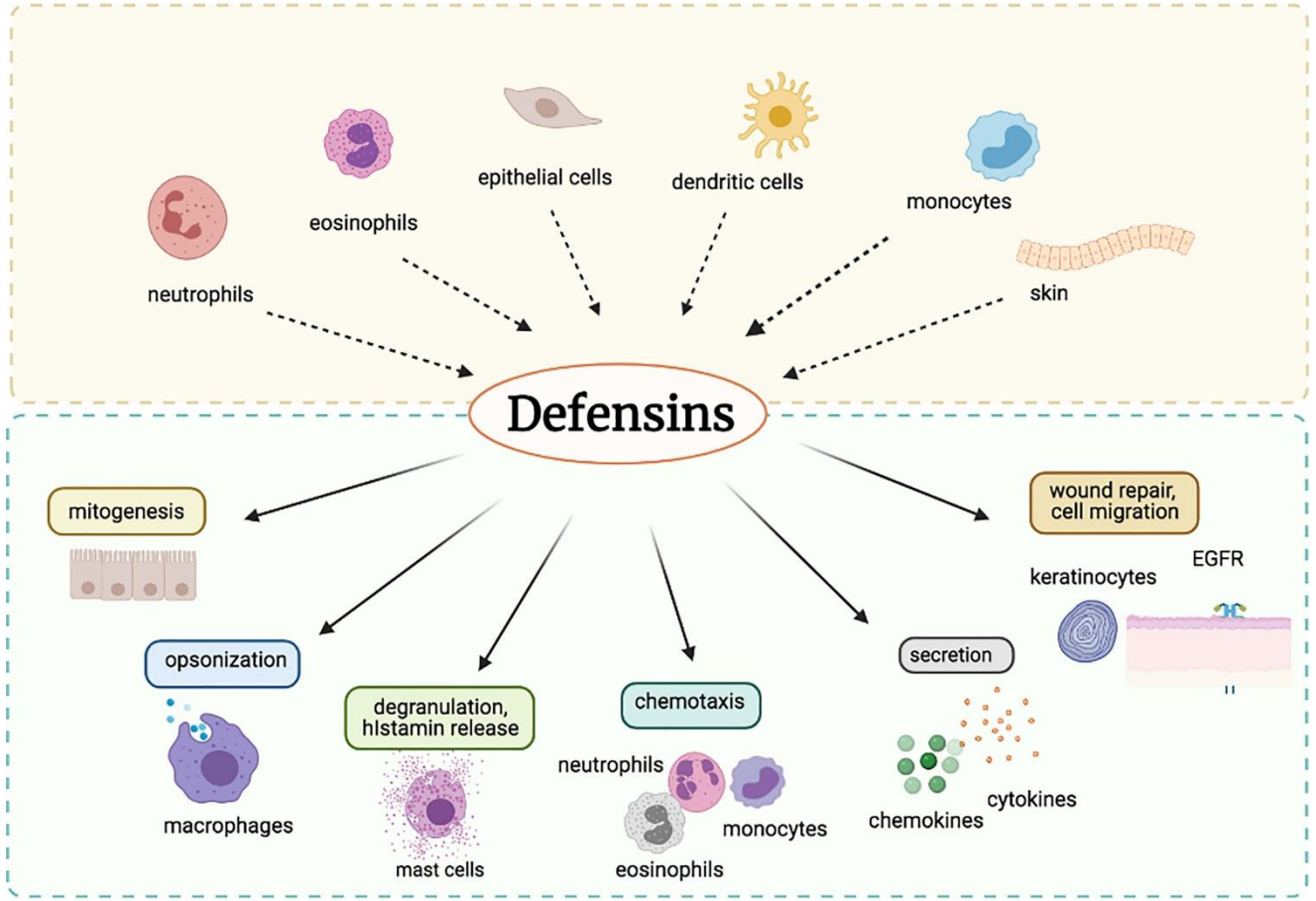
$$f(x)_{id} = \frac{0.01a}{0.01 + e^{(-bx)}},$$

$$f(x)_{vl} = \frac{a}{0.01 + e^{x}},$$

nsSNPs in DEFB1 gene reveal impact on protein-ligand binding sites

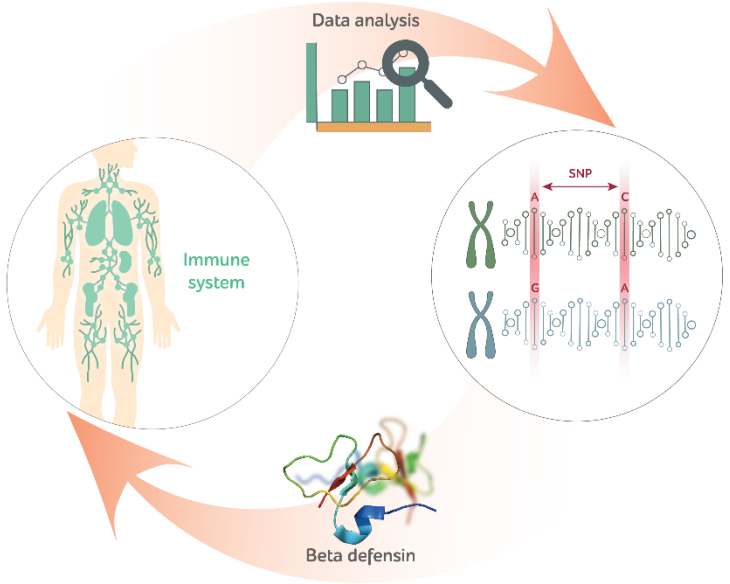


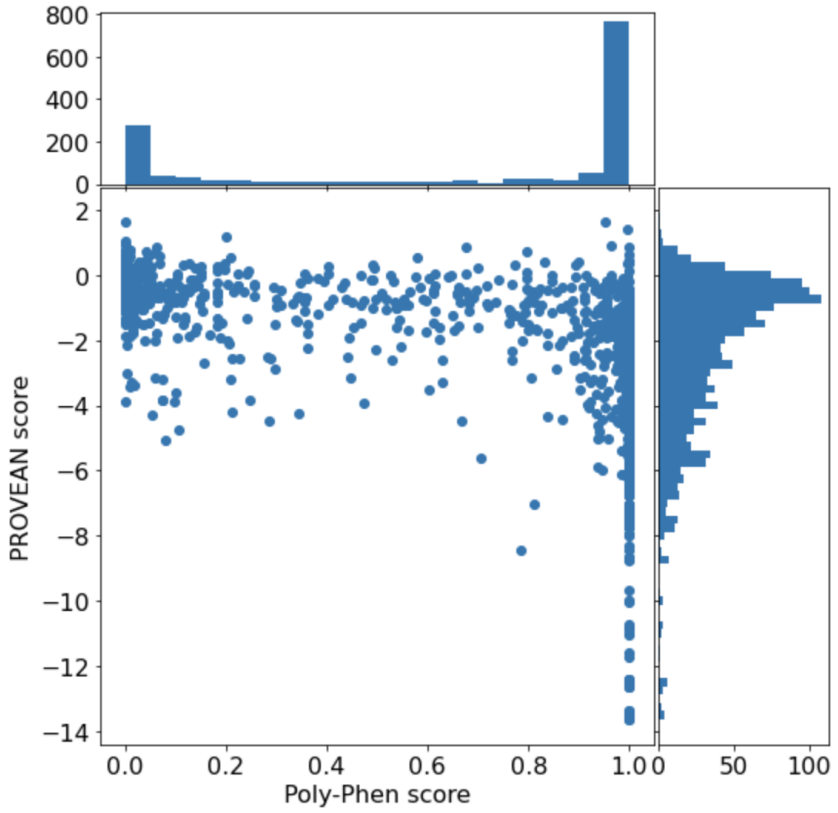


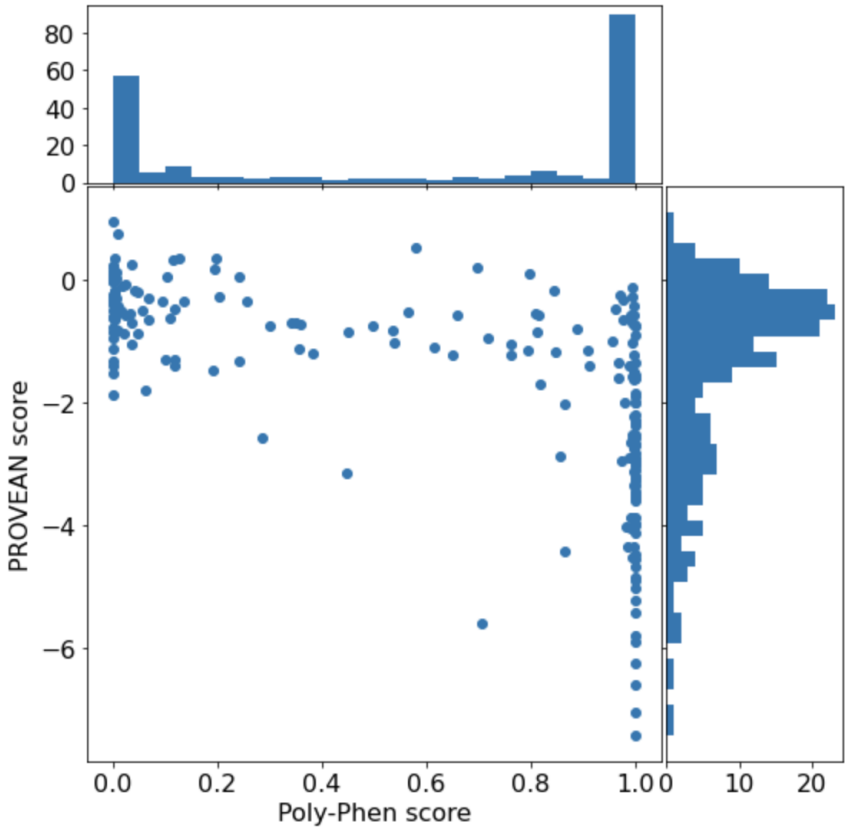


Fareed et al., Comp. Biol .Chem., 2022

nsSNPs in DEFB1 gene reveal impact on protein-ligand binding sites

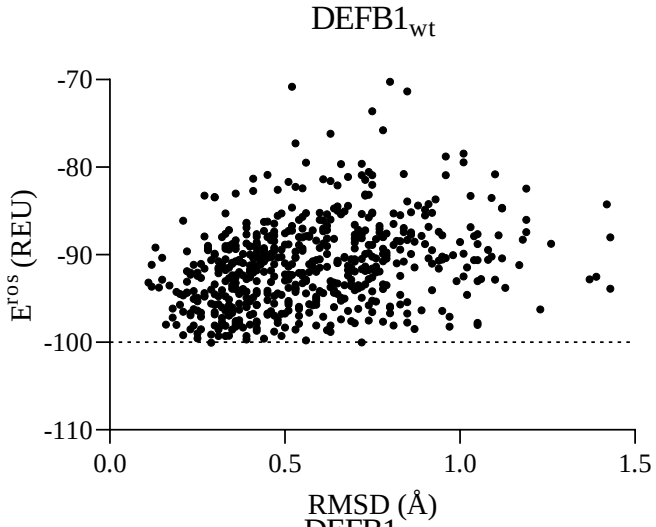


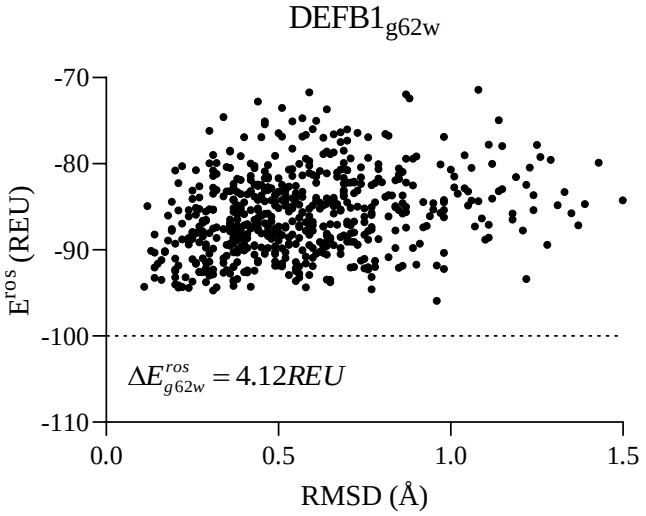


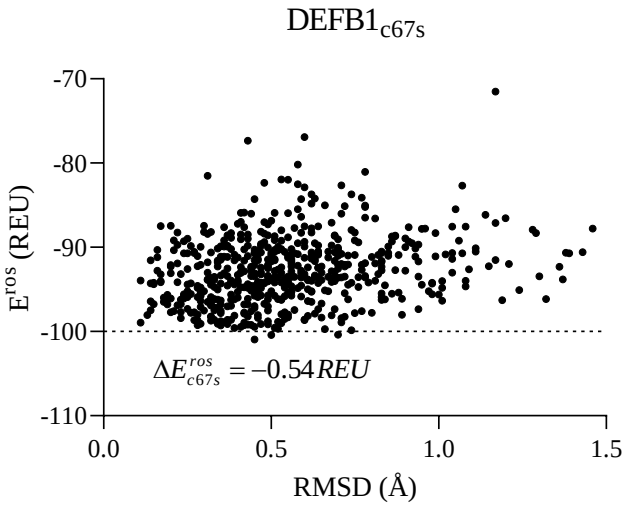


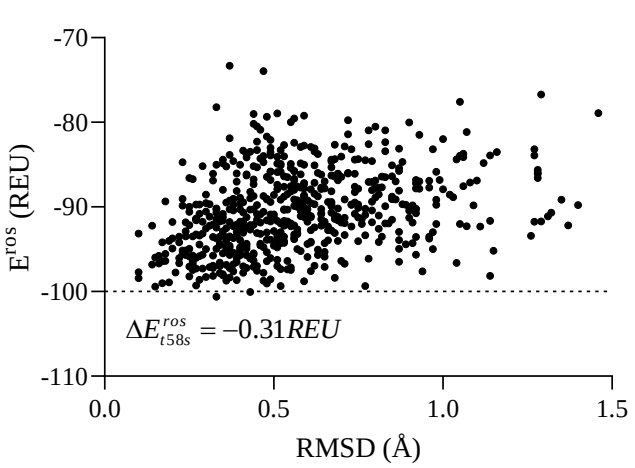
Fareed et al., Comp. Biol .Chem., 2022

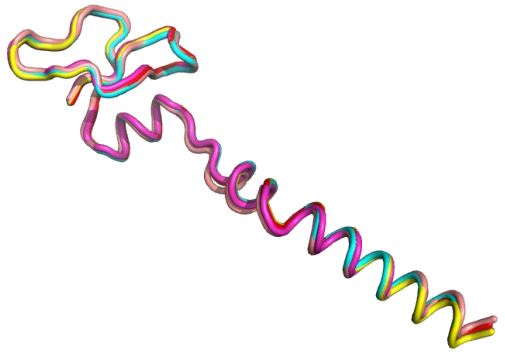
nsSNPs in DEFB1 gene reveal impact on protein-ligand binding sites





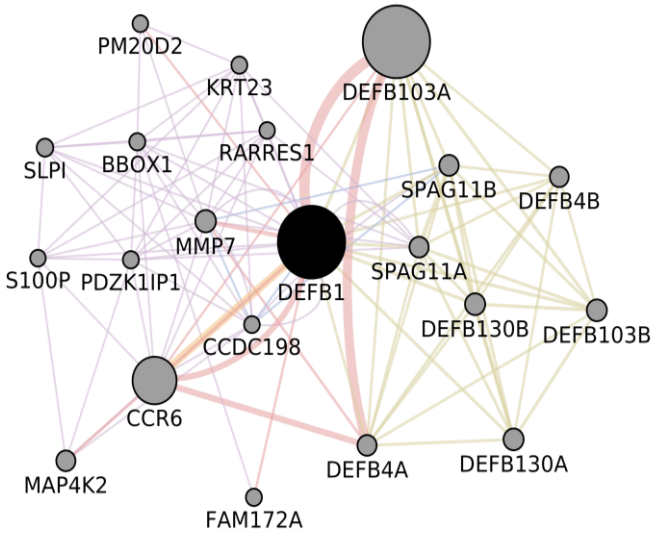


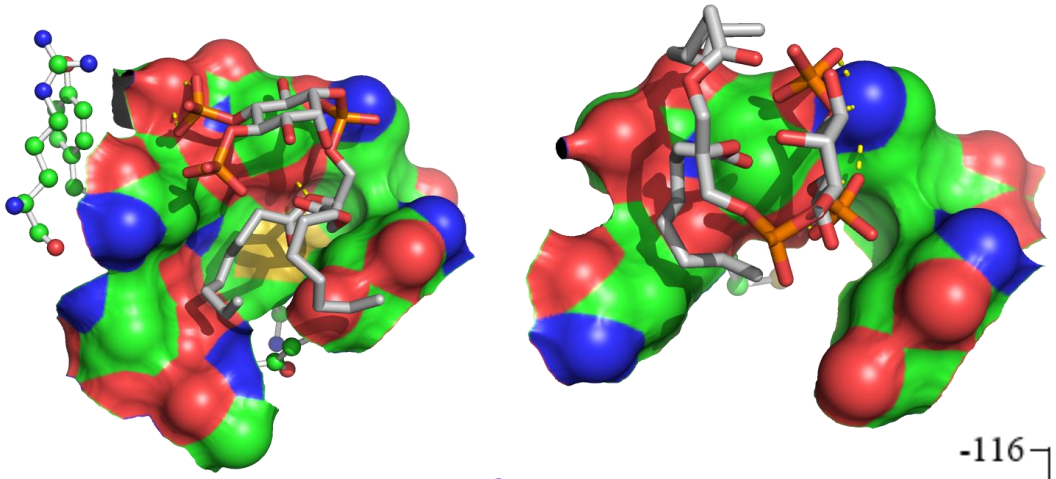


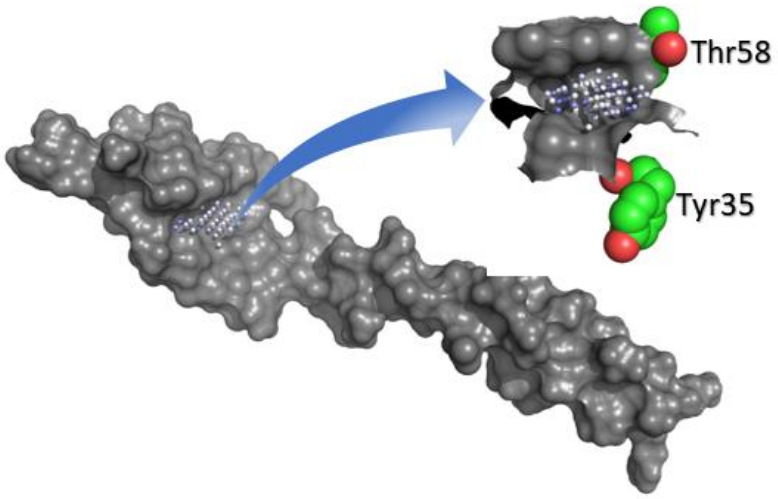


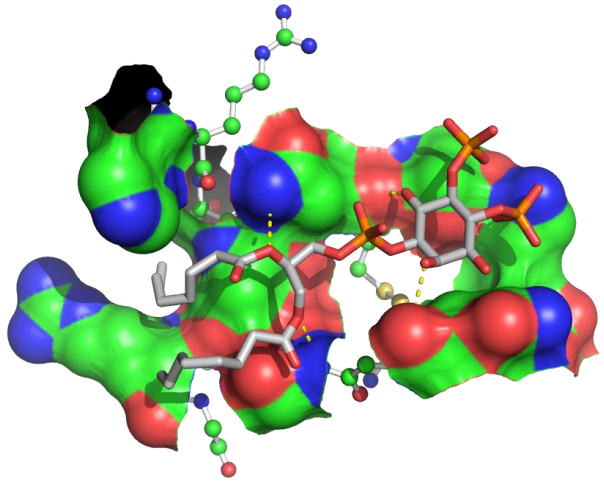
Fareed et al., Comp. Biol .Chem., 2022

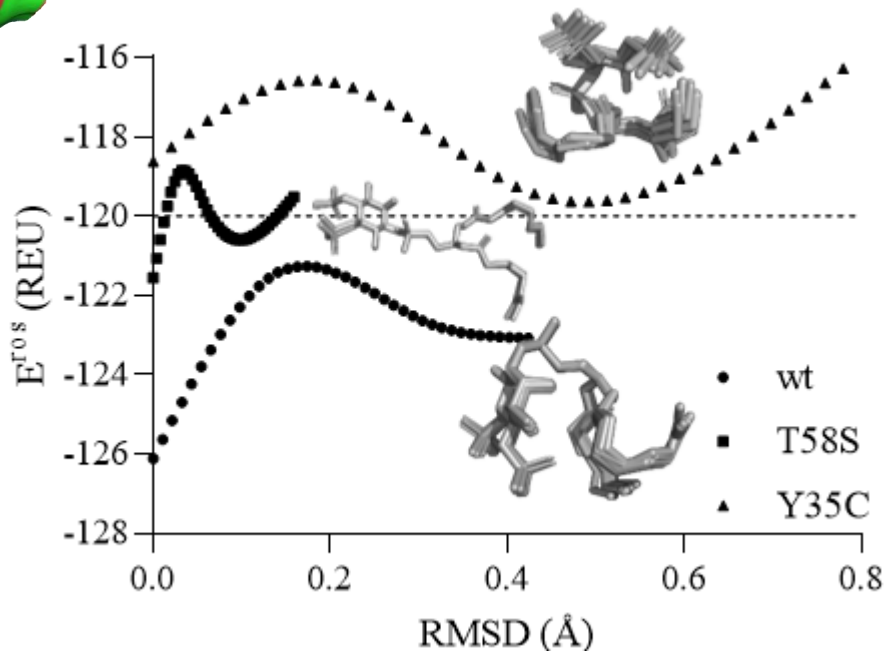
nsSNPs in DEFB1 gene reveal impact on protein-ligand binding sites





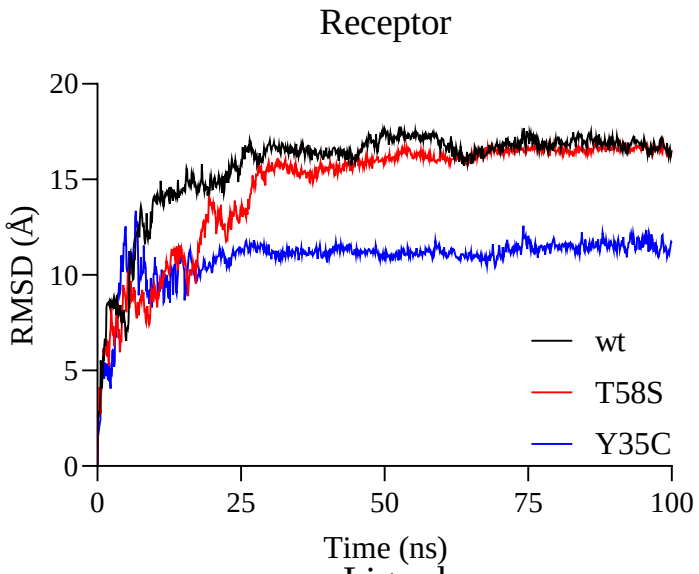


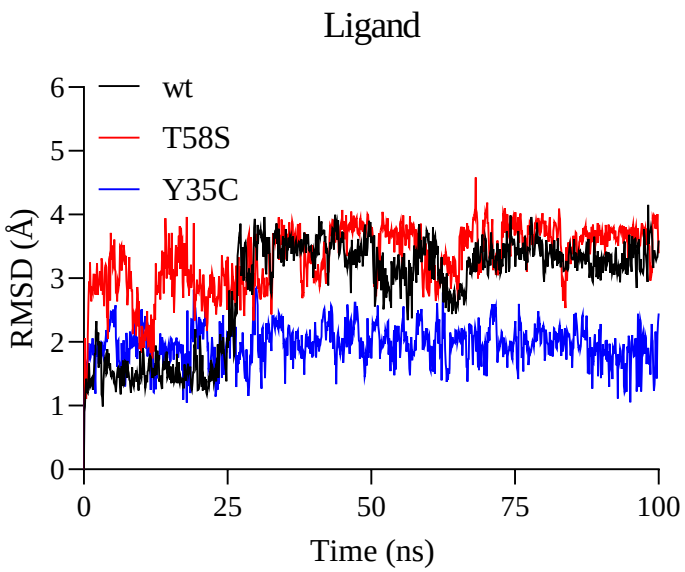


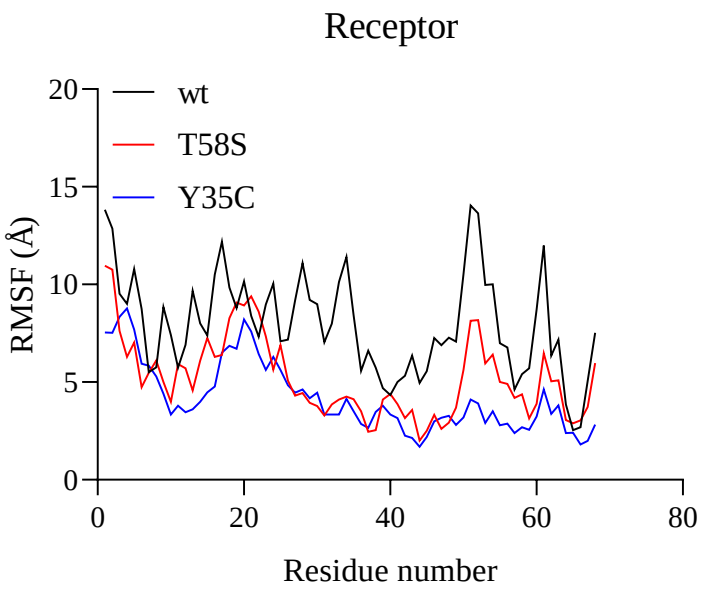


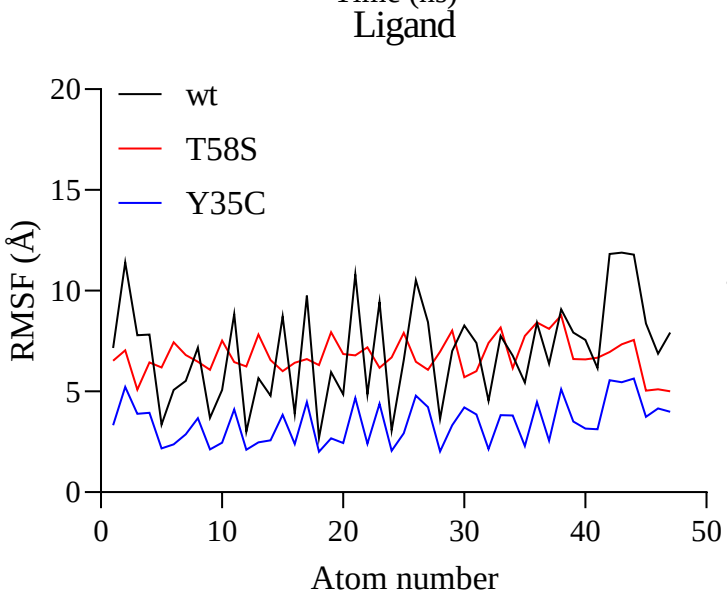
Fareed et al., Comp. Biol .Chem., 2022

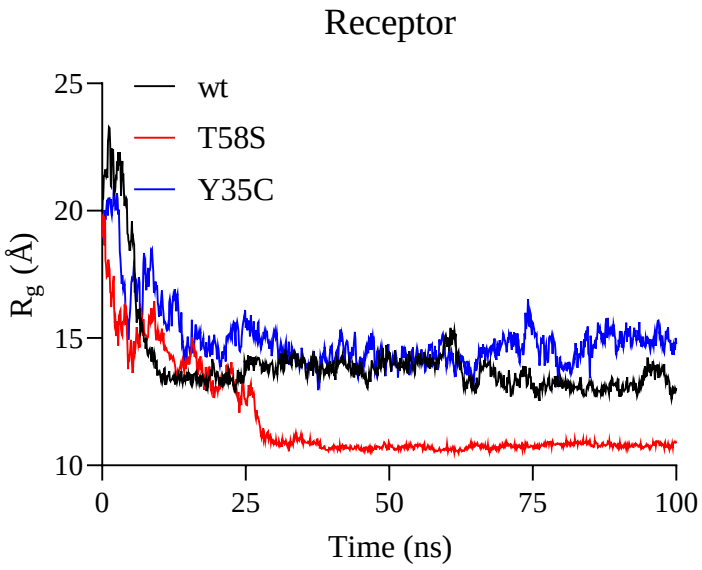
nsSNPs in DEFB1 gene reveal impact on protein-ligand binding sites

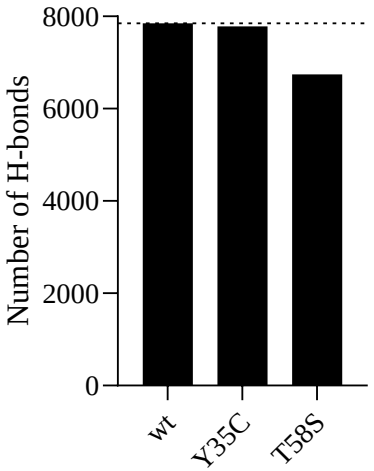






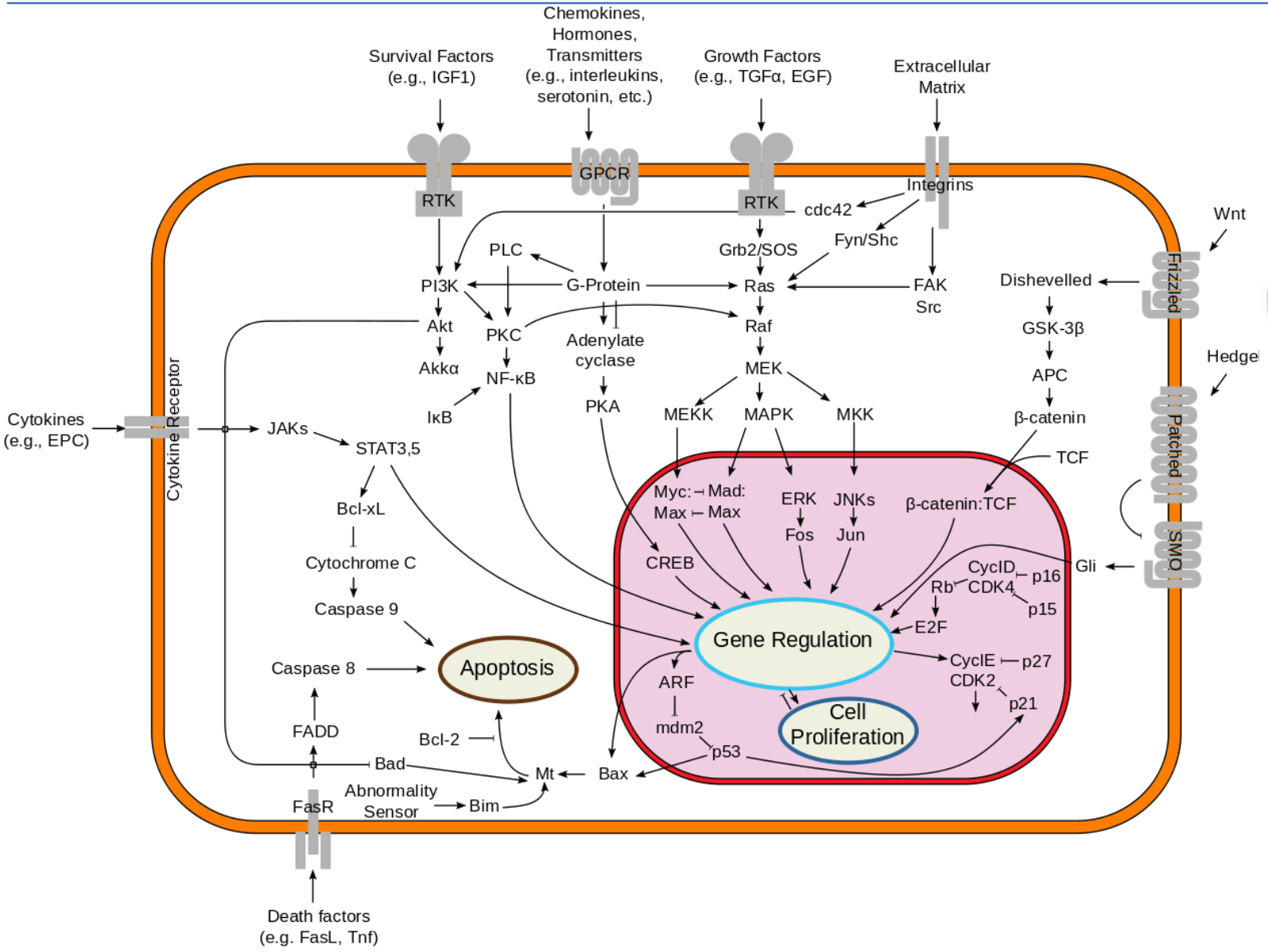


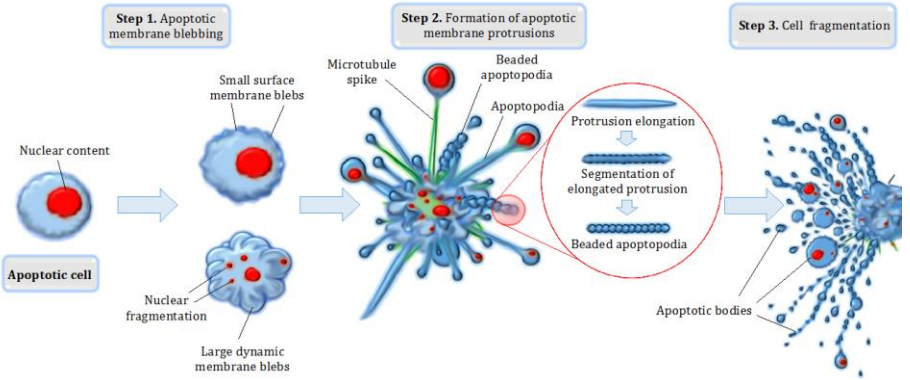


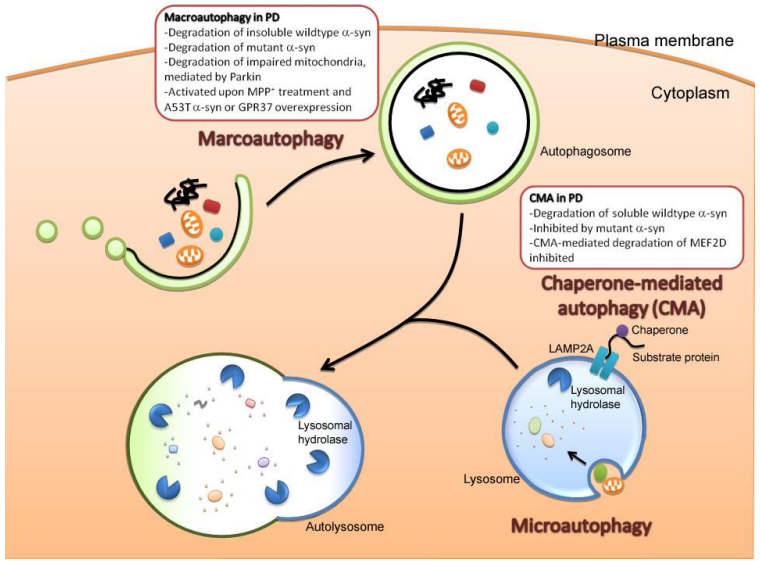


Fareed et al., Comp. Biol .Chem., 2022

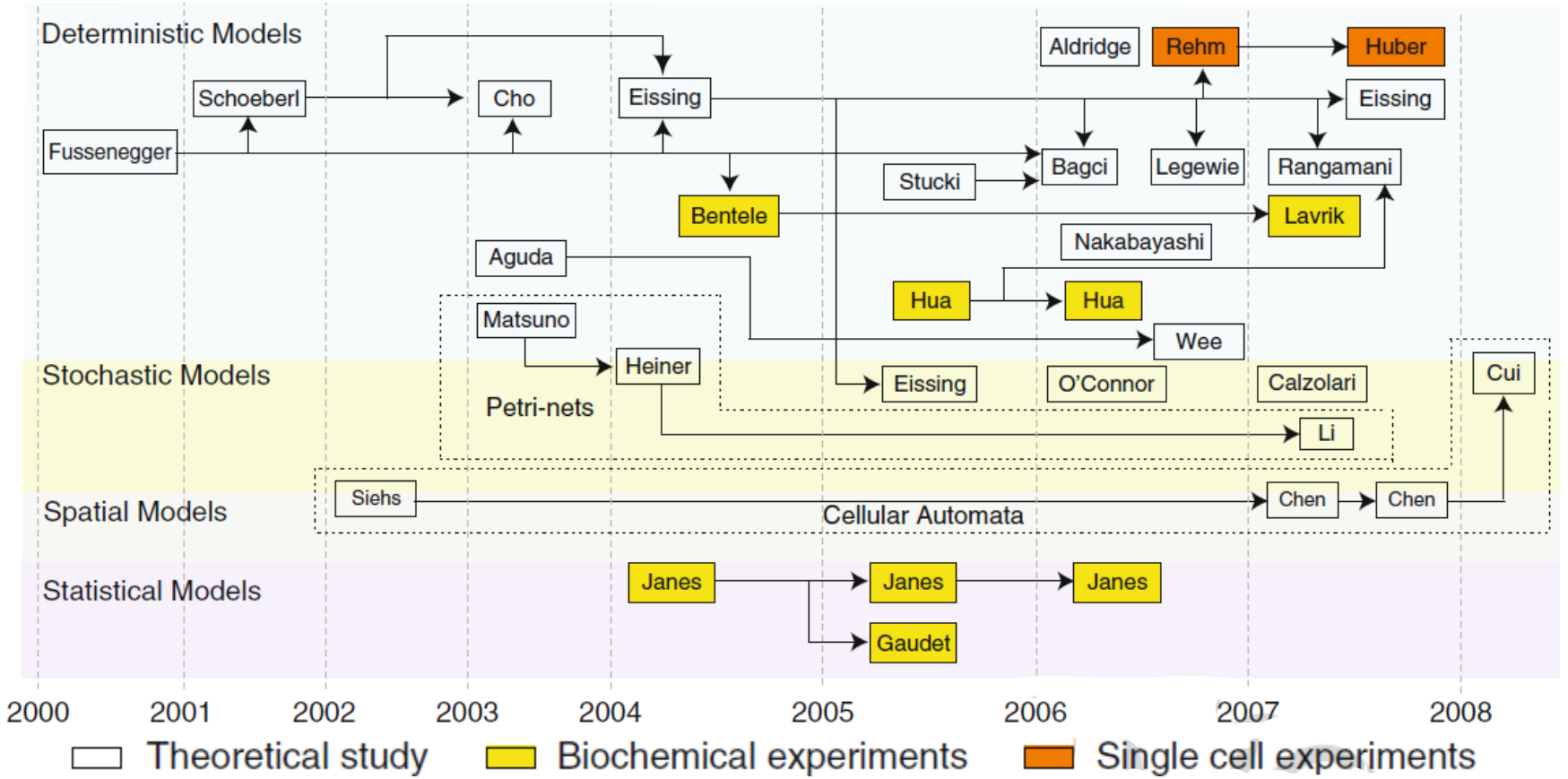
Cell death







Cell death modelling



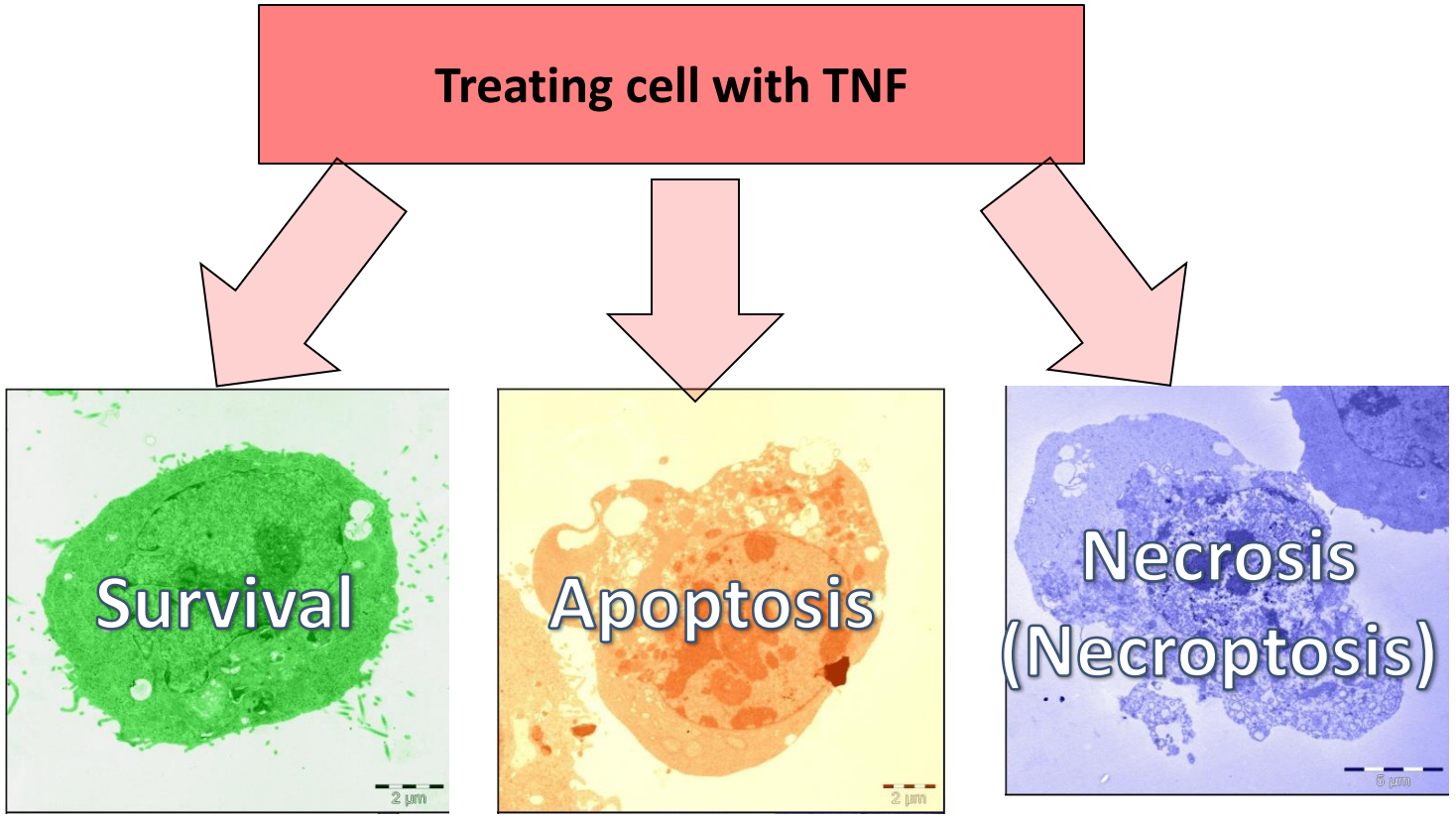
(From Huber, Bullinger and Rehm, Systems Biology Approaches to the Study of Apoptosis 2009)

Cell death modelling

Cell Death Modalities

(From Galuzzi et al, Cell Death and Diff, 2007)

Apoptosis vs. Necrosis vs. Survival



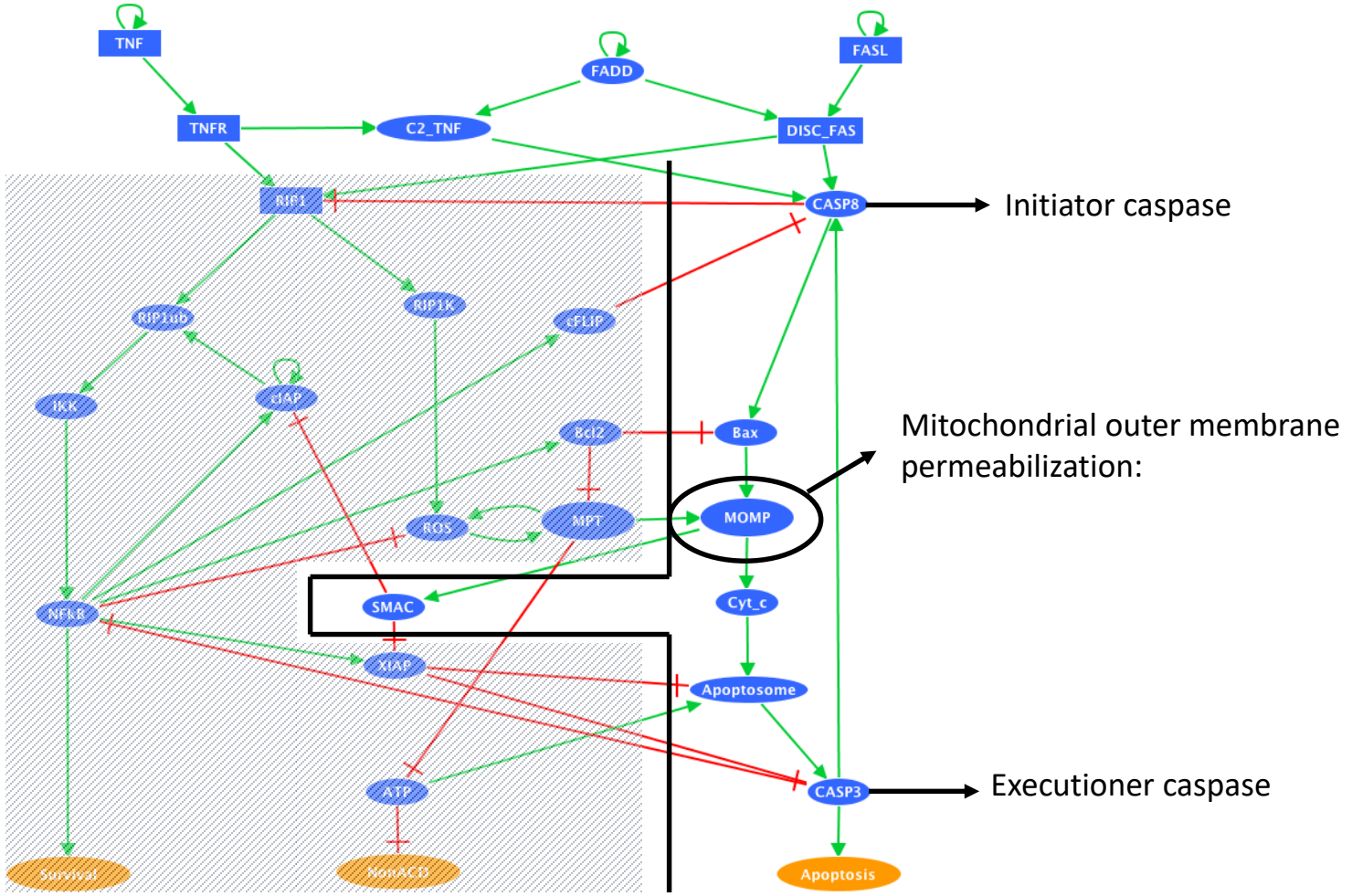
OPEN ACCESS Freely available online
PLoS COMPUTATIONAL BIOLOGY

Mathematical Modelling of Cell-Fate Decision in Response to Death Receptor Engagement

Laurence Calzone[1,2,3]*, Laurent Tournier[1,2,3], Simon Fourquet[1,2,3], Denis Thieffry[4,5], Boris Zhivotovsky[6], Emmanuel Barillot[1,2,3¶], Andrei Zinovyev[1,2,3¶]

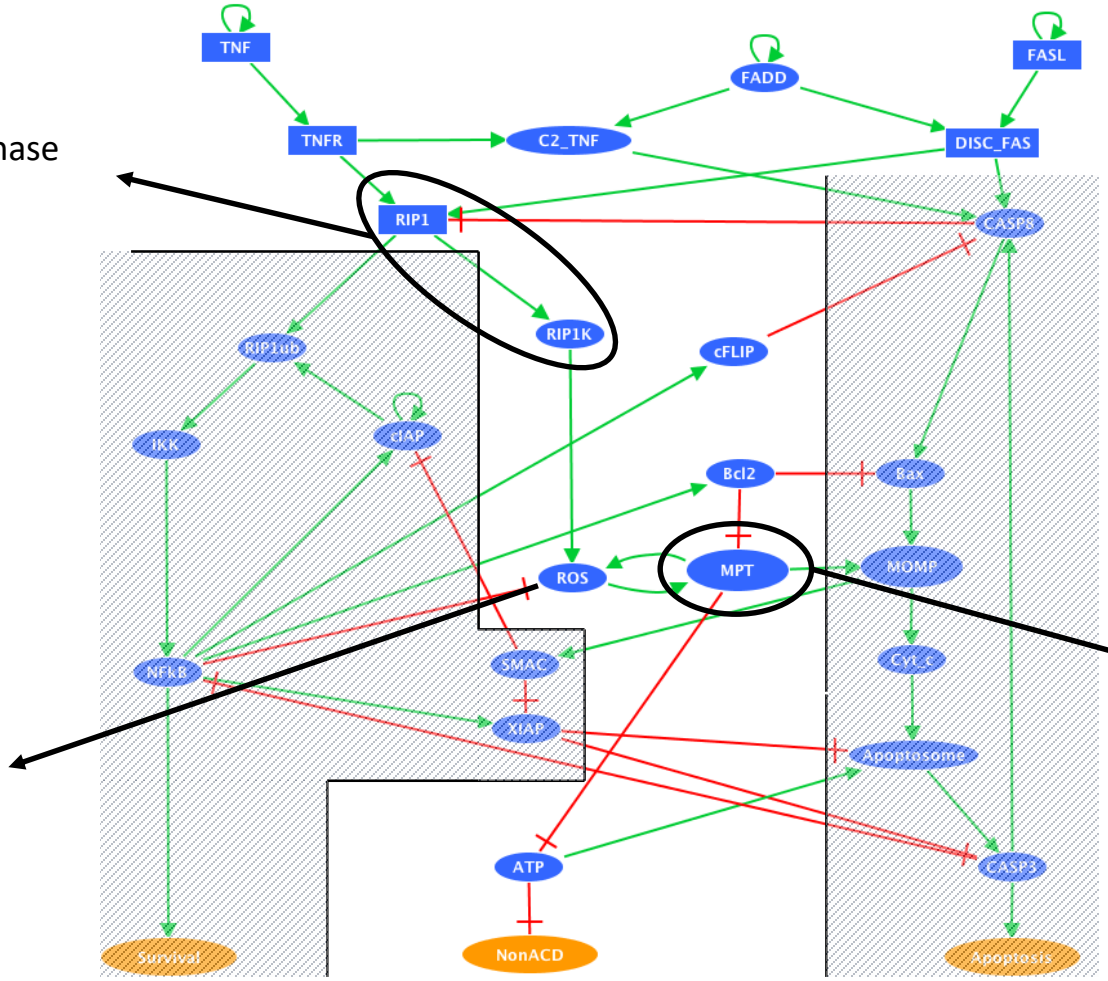
1 Institut Curie, Paris, France, 2 Ecole des Mines ParisTech, Paris, France, 3 INSERM U900, Paris, France, 4 TAGC – INSERM U928 & Université de la Méditerranée, Marseille, France, 5 CONTRAINTES Project, INRIA Paris-Rocquencourt, France, 6 Karolinska Institutet, Stockholm, Sweden

Apoptosis


TNF
FADD
FASL
TNFR
C2_TNF
DISC_FAS
RIP1
CASP8
Initiator caspase
RIP1ub
RIP1K
cFLIP
IKK
cIAP
Bcl2
Bax
Mitochondrial outer membrane permeabilization:
ROS
MPT
MOMP
SMAC
Cyt_c
NFkB
XIAP
Apoptosome
ATP
CASP3
Executioner caspase
Survival
NonACD
Apoptosis

Necrosis

Necrosis needs kinase activity of RIP1



Mitochondria Permeability Transition

ROS : Reactive Oxygen Species

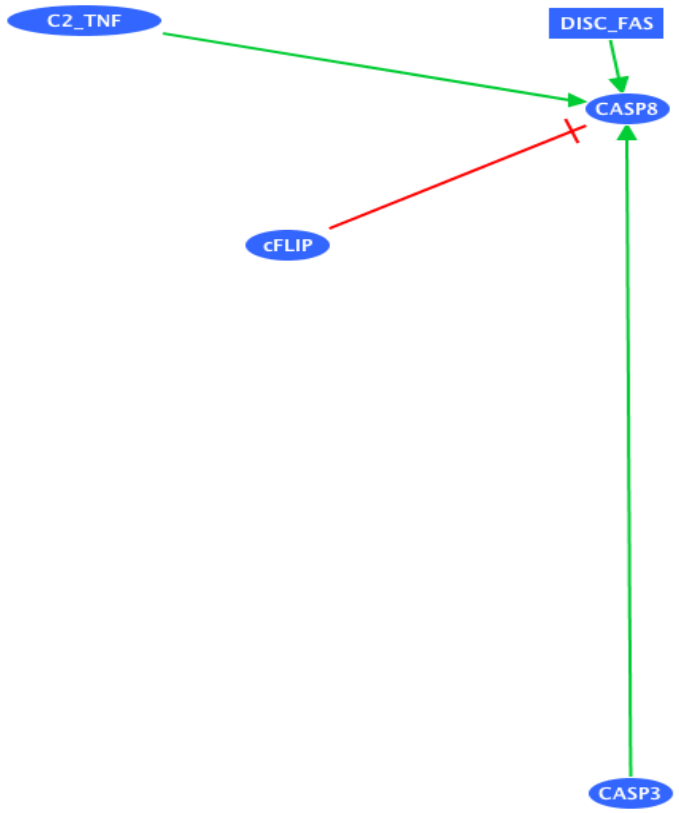
Boolean modeling

Назначить булеву функцию на узел

Example of CASP8

CASP8 = 0 when
DISC-Fas=0 and DISC-TNF=0 and CASP3=0
(equivalent to no external signals from death receptors
and no intracellular problems)
cFLIP=1
(equivalent to inhibition by the NFkB pathway)

CASP8 = 1 when
DISC-Fas=1 or/and DISC-TNF=1
(equivalent to signal from death receptors)
CASP3=1
(amplification signal, feedback activation)
AND no cFLIP



One node = one species

Target molecules in various tumors

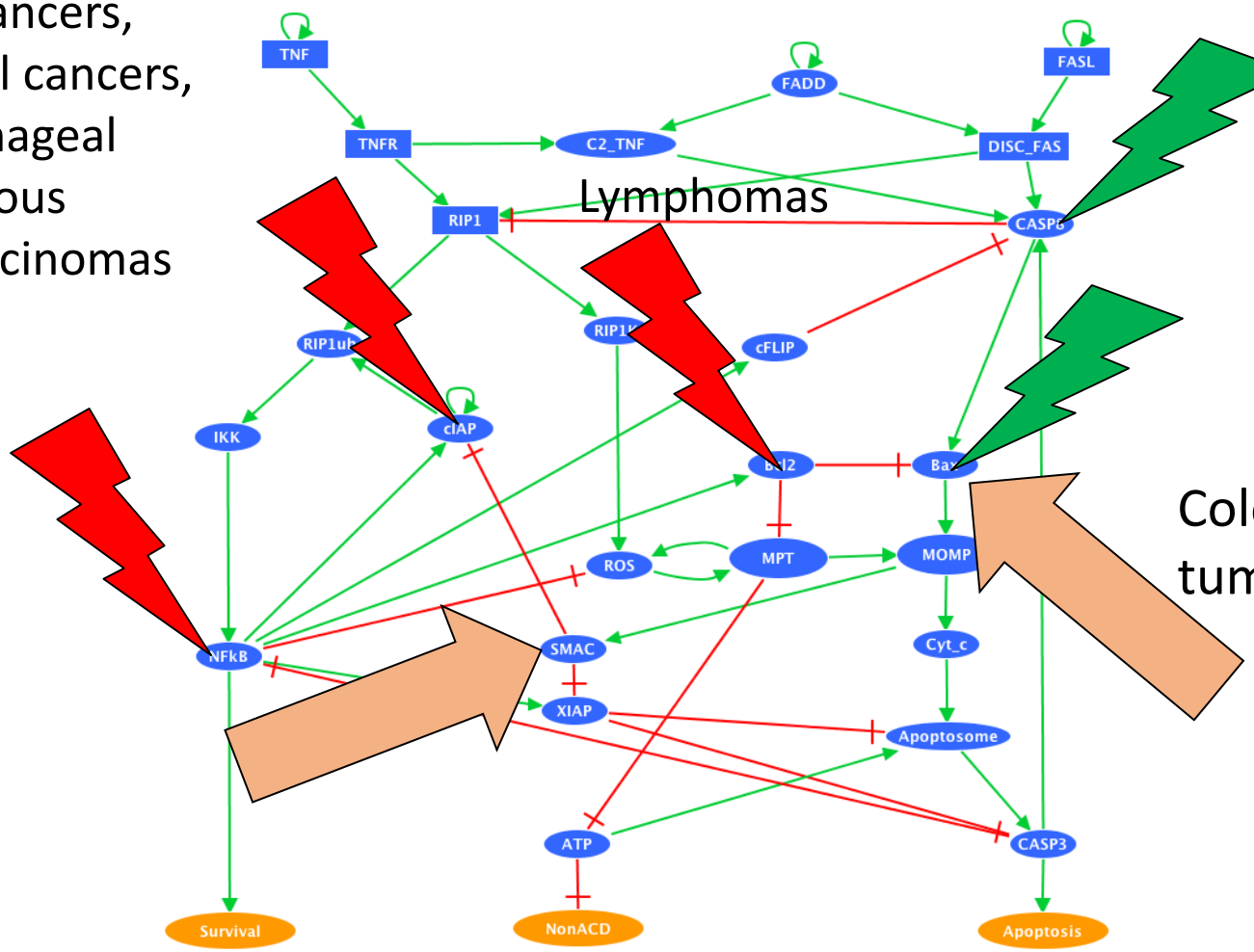
Lung cancers,
cervical cancers,
oesophageal
squamous
cell carcinomas

Lymphomas

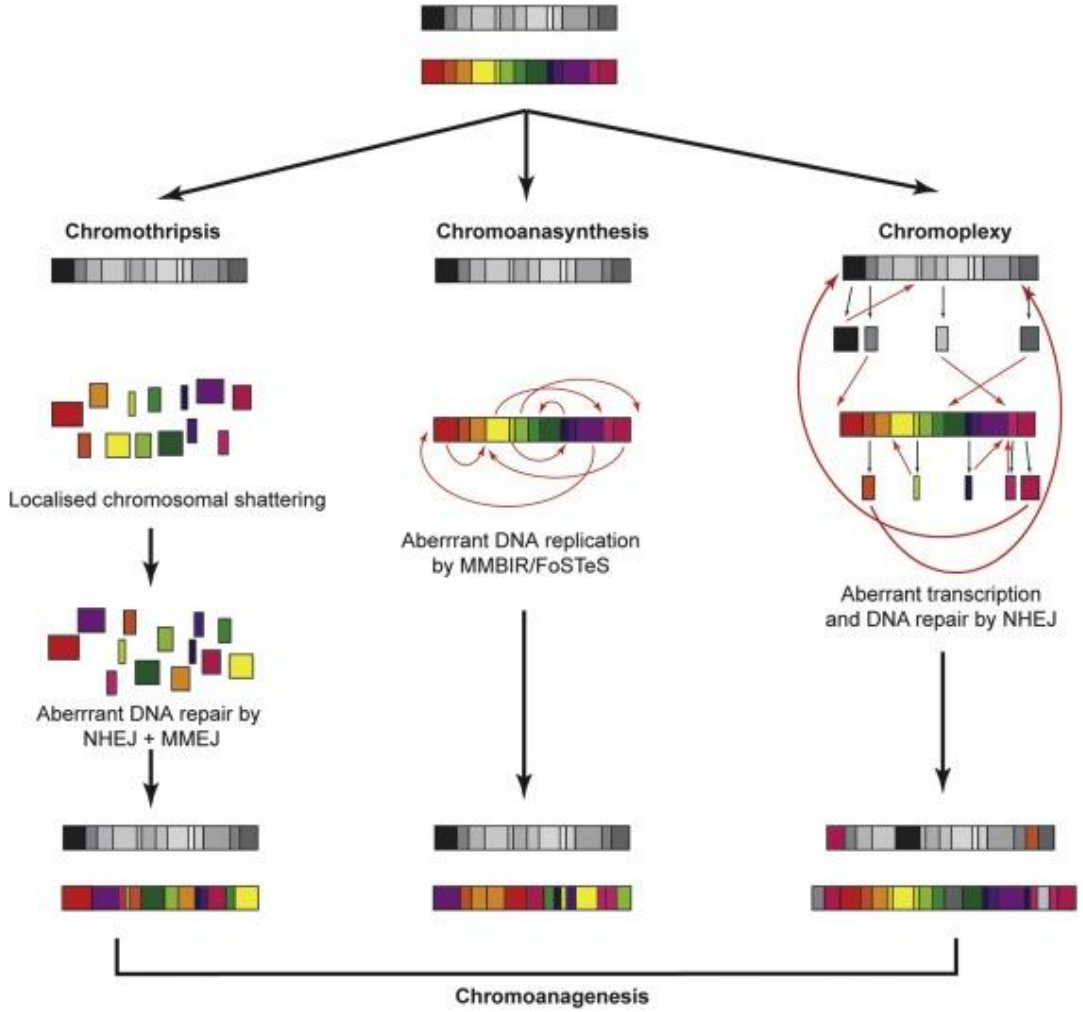
Ewing's
sarcoma,
lung cancer,
neuroblastomas

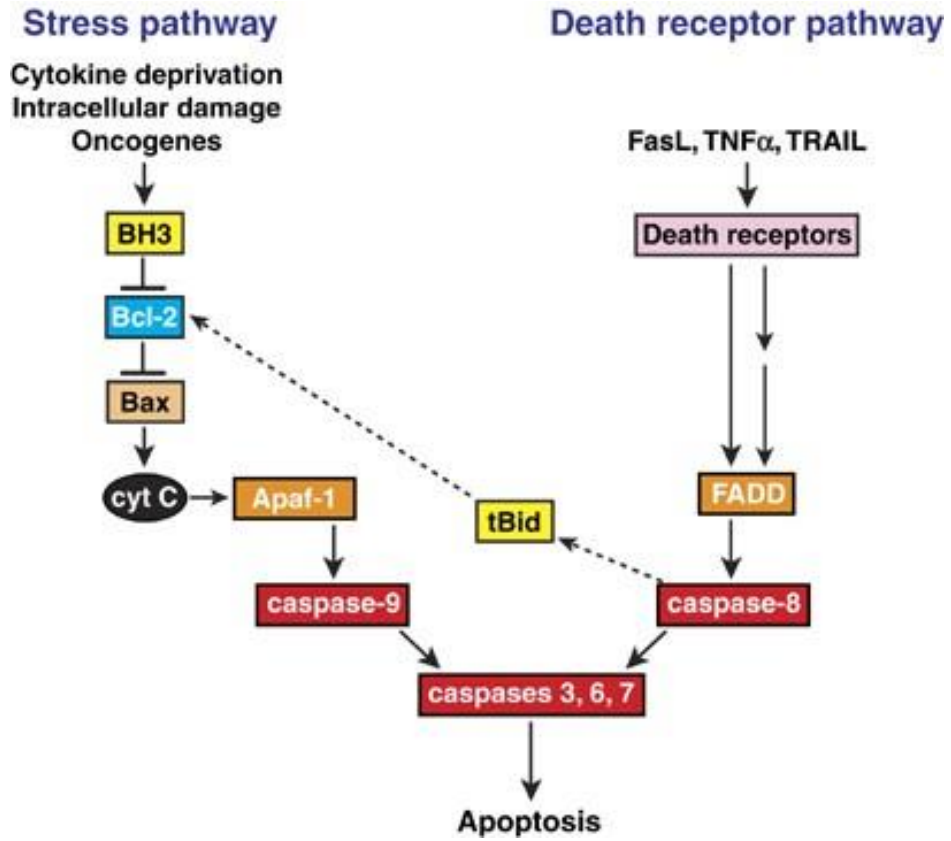
Lymphomas,
breast cancer

Colorectal
tumors

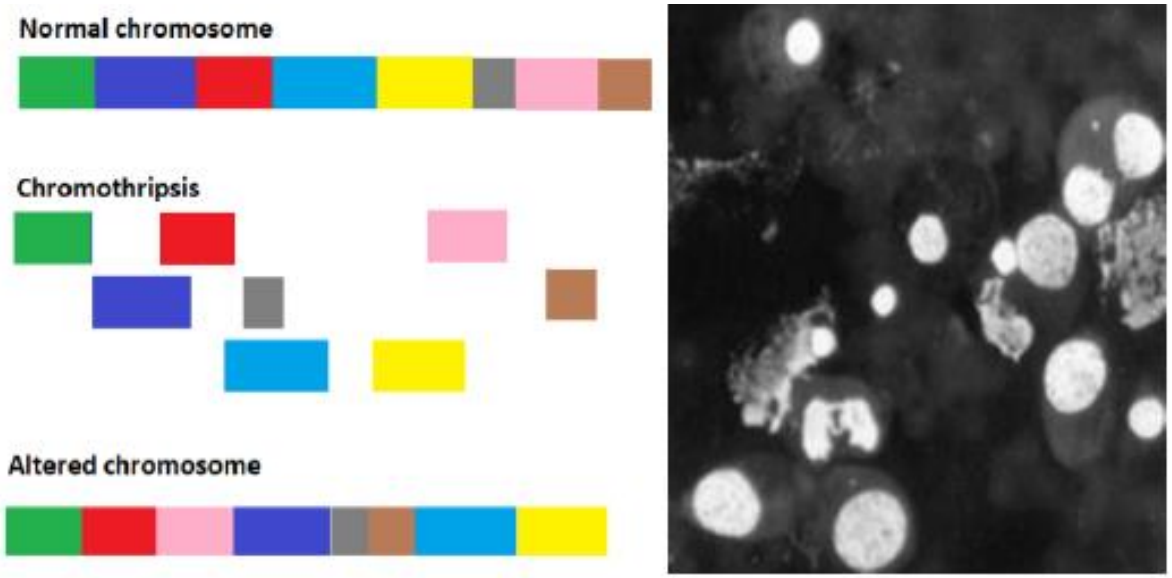


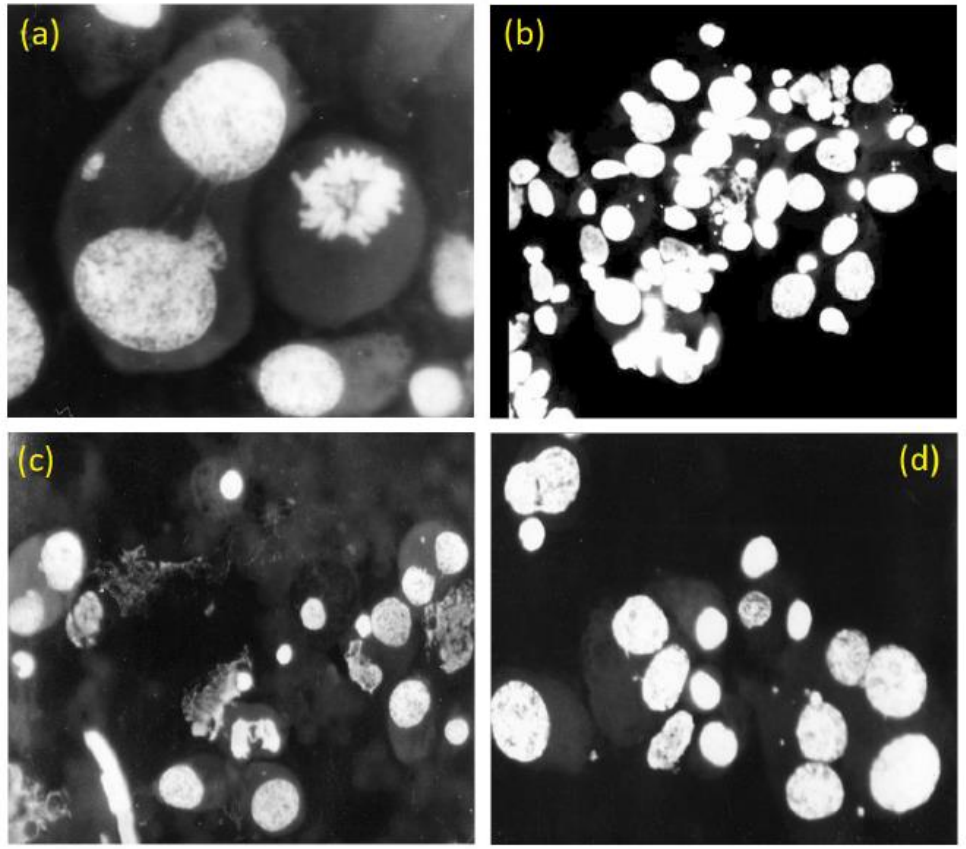
Genome chaos





Genome chaos





Genome chaos algorithm

```
gene_sequence = list("ATGGCGCACGCTGGGAGAACAGGGTACGATA

# Calculate the number of nucleotides
num_nucleotides = len(gene_sequence)

print("Number of Nucleotides:", num_nucleotides)
```

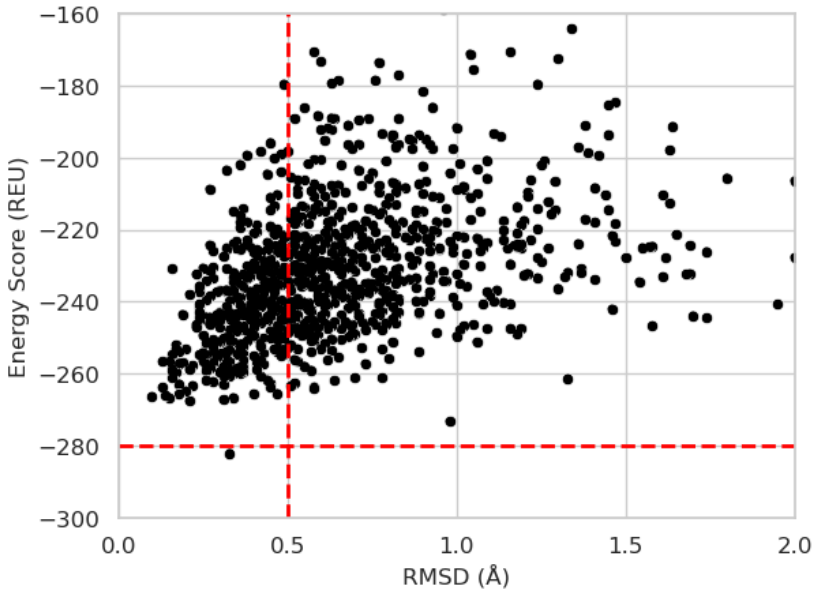
```
import random

# Mutation rate
mutation_rate = 1.0e-4

# Number of cell divisions
num_divisions = 60

# Initialize the gene sequence (as a list of characters, for example)
gene_sequence = list("ATGGCGCACGCTGGGAGAACAGGGTACGATAACCGGGAGATAGTGATGAAGTACA

# Function to apply mutations
def apply_mutation(gene_sequence, mutation_rate):
    mutated_sequence = []
    for base in gene_sequence:
        if random.random() < mutation_rate:
            # Mutate the base (for simplicity, just change to a random base)
            mutated_base = random.choice("ATCG")
            mutated_sequence.append(mutated_base)
        else:
            mutated_sequence.append(base)
    return mutated_sequence

# Simulate cell divisions and mutations
for _ in range(num_divisions):
    gene_sequence = apply_mutation(gene_sequence, mutation_rate)

# Print the mutated gene sequence
mutated_gene = ''.join(gene_sequence)
print("Mutated Gene Sequence:", mutated_gene)
```

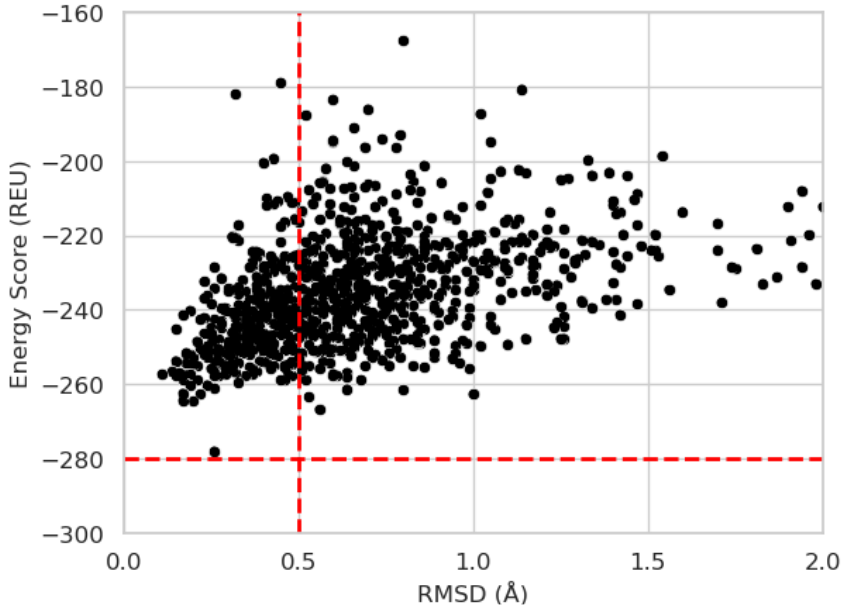
```
# Genetic code dictionary
genetic_code = {
    "TTT": "F", "TTC": "F", "TTA": "L", "TTG": "L",
    "CTT": "L", "CTC": "L", "CTA": "L", "CTG": "L",
    "ATT": "I", "ATC": "I", "ATA": "I", "ATG": "M",
    "GTT": "V", "GTC": "V", "GTA": "V", "GTG": "V",
    "TCT": "S", "TCC": "S", "TCA": "S", "TCG": "S",
    "CCT": "P", "CCC": "P", "CCA": "P", "CCG": "P",
    "ACT": "T", "ACC": "T", "ACA": "T", "ACG": "T",
    "GCT": "A", "GCC": "A", "GCA": "A", "GCG": "A",
    "TAT": "Y", "TAC": "Y", "TAA": "*", "TAG": "*",
    "CAT": "H", "CAC": "H", "CAA": "Q", "CAG": "Q",
    "AAT": "N", "AAC": "N", "AAA": "K", "AAG": "K",
    "GAT": "D", "GAC": "D", "GAA": "E", "GAG": "E",
    "TGT": "C", "TGC": "C", "TGA": "*", "TGG": "W",
    "CGT": "R", "CGC": "R", "CGA": "R", "CGG": "R",
    "AGT": "S", "AGC": "S", "AGA": "R", "AGG": "R",
    "GGT": "G", "GGC": "G", "GGA": "G", "GGG": "G",
}

# Function to translate a DNA sequence to a protein sequence
def translate_dna_to_protein(dna_sequence):
    protein_sequence = []
    for i in range(0, len(dna_sequence), 3):
        codon = dna_sequence[i:i + 3]
        amino_acid = genetic_code.get(codon, 'X')  # 'X' for unknown codons
        protein_sequence.append(amino_acid)
    return ''.join(protein_sequence)

# Translate the reference and mutated sequences
translated_reference_sequence = translate_dna_to_protein(reference_sequence)
translated_mutated_sequence = translate_dna_to_protein(mutated_gene)

print("Translated Ref Gene Sequence:", translated_reference_sequence)
print("Translated Mut Gene Sequence:", translated_mutated_sequence)
```

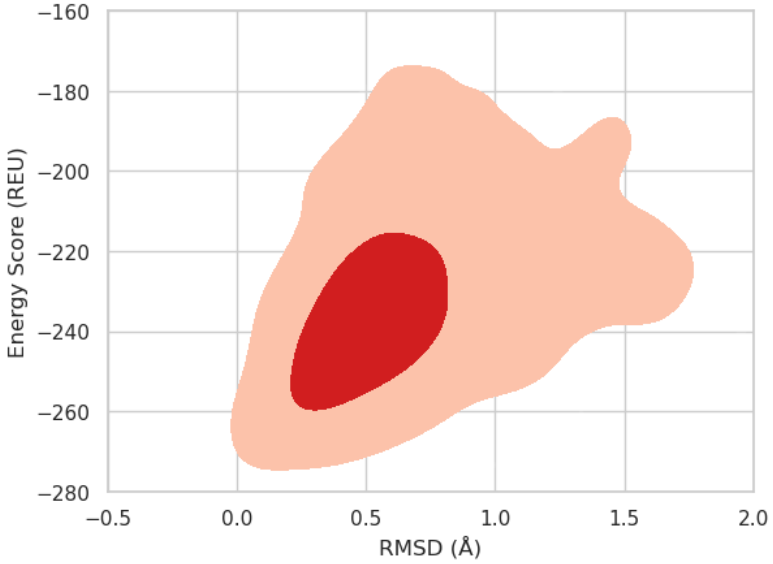
```
Protein Sequence 1: MAHAGRTGYDNREIVMKYIHYI
LTLRQAGDDFSRRYRRDFAEMSSQLHLTPFTARGRFATVVEI
LFDFSWLSLKTLLSLALVGACITLGAYLGHK*
Protein Sequence 2: MAHAWRTGYDNREIVMKYIHYI
LTLRQAGDDFSRRYRRDFAEMSSQLHLTPFTARGRFATVVEI
LFDFSWLSLKTLLSLALVGACITLGAYLGHK*
Sequence Identity: 99.58%
```

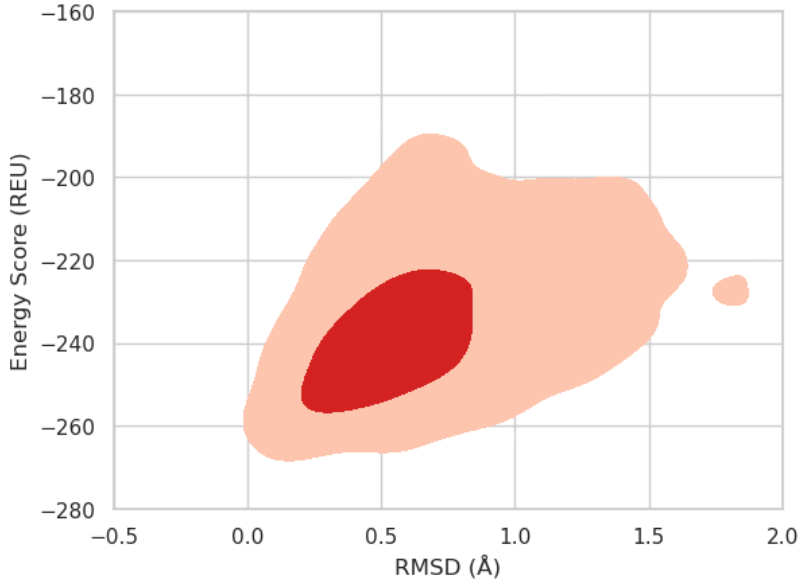
ITMOre than a UNIVERSITY

Genome chaos

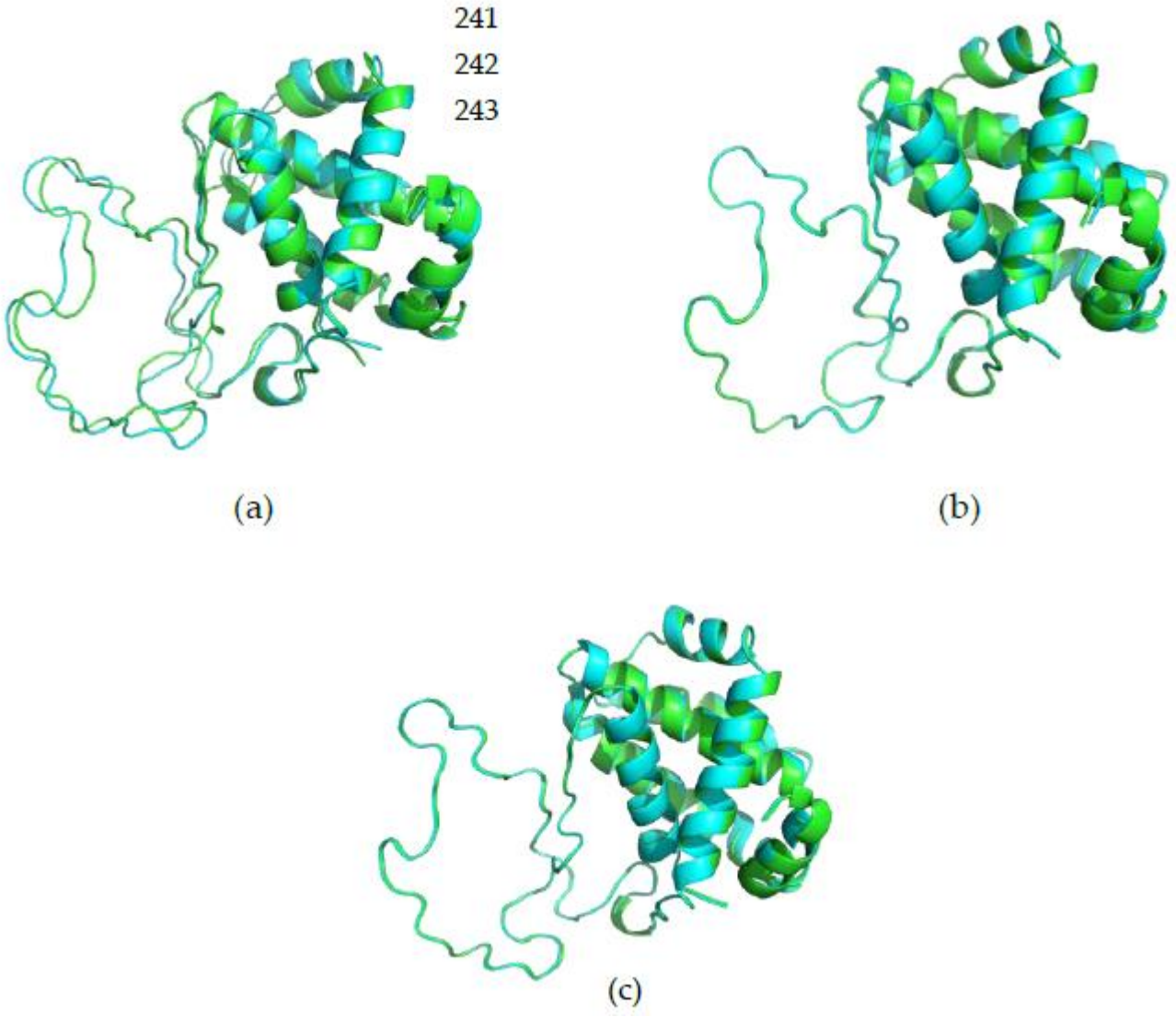


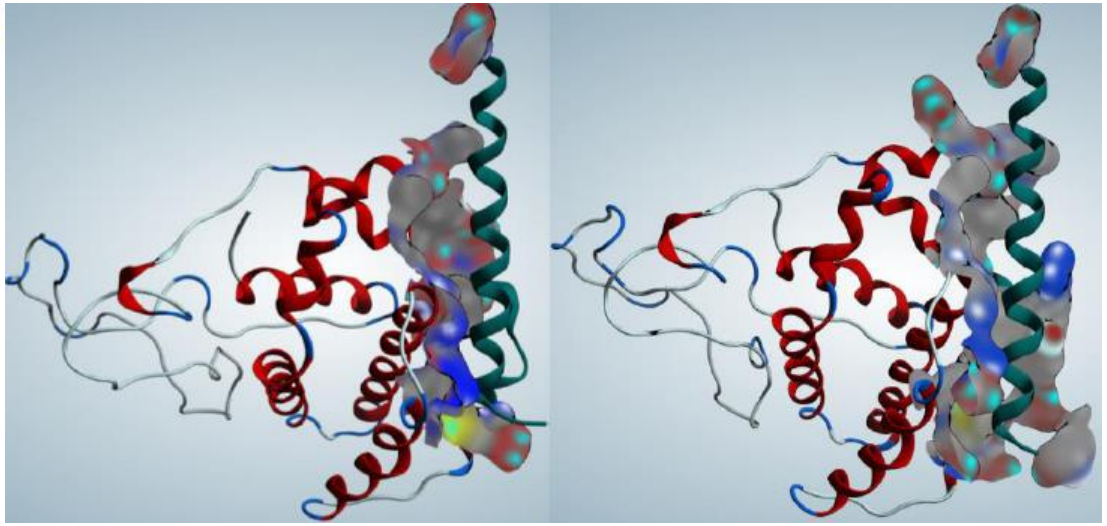






Genome chaos (BCL-2/BH3 interaction)

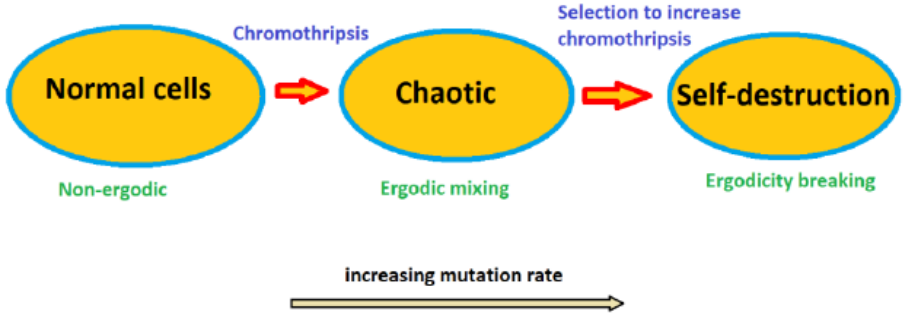


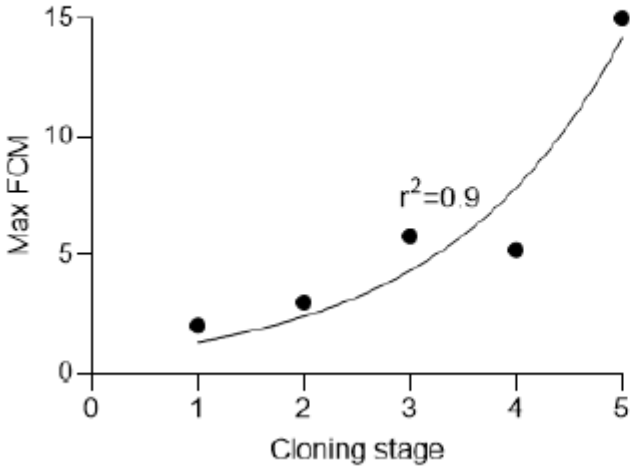


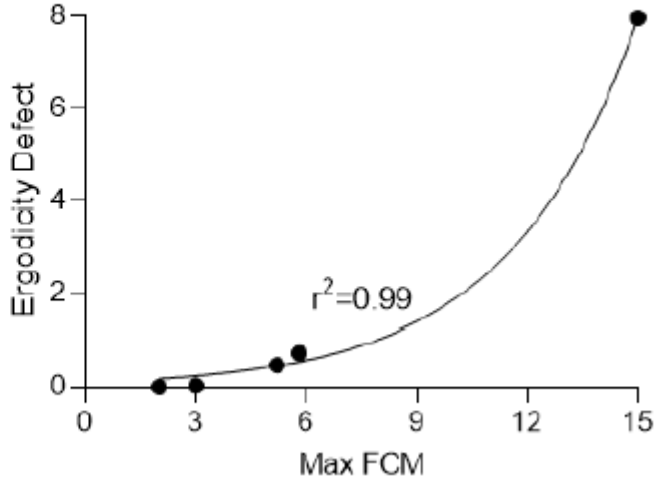
Genome chaos

Table 1. Changes of observed minimum, average, and maximum FCM and calculated effective mutation rate and ergodicity defect at different stages of the selection for increasing FCM.

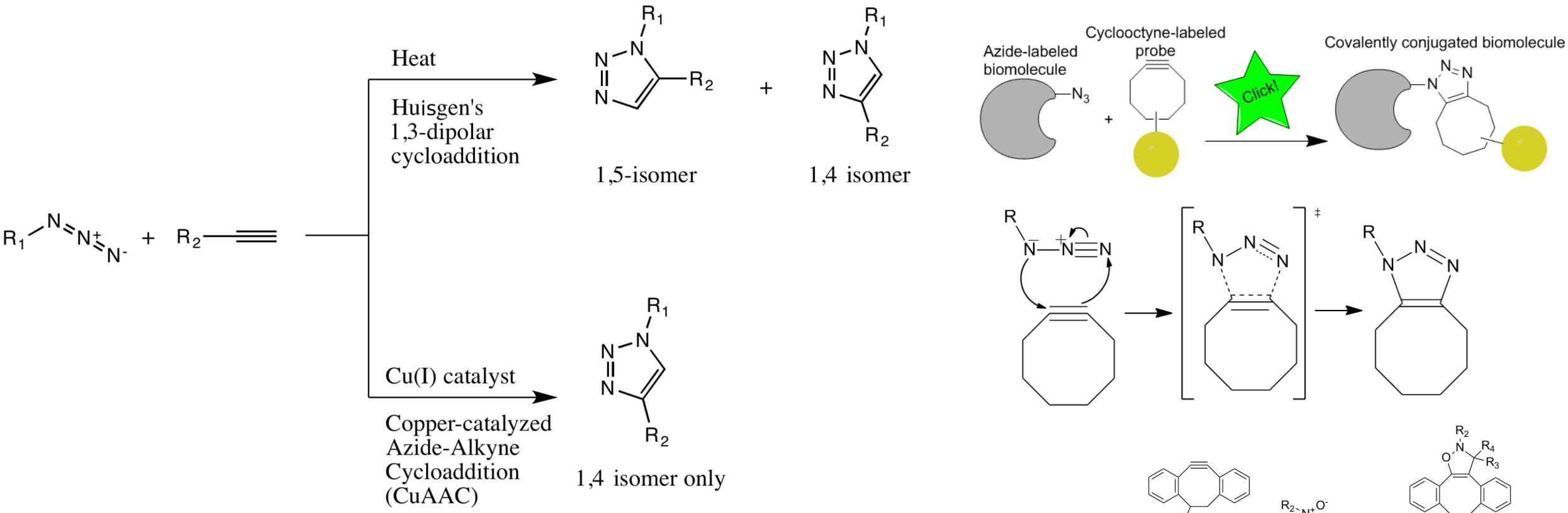
Stage	Clones	Min FCM	Average FCM	Max FCM	Effective mutation rate, $\times 10^{-4}$	Ergodicity defect
0	48	0.0	0.6	2.0	6.99	0
1	50	0.0	0.8	3.0	8.99	0.03
2	52	0.0	1.7	5.8	17.98	0.74
3	48	0.0	1.6	5.2	16.98	0.48
4	47	1.0	4.7	15.0	47.95	7.96

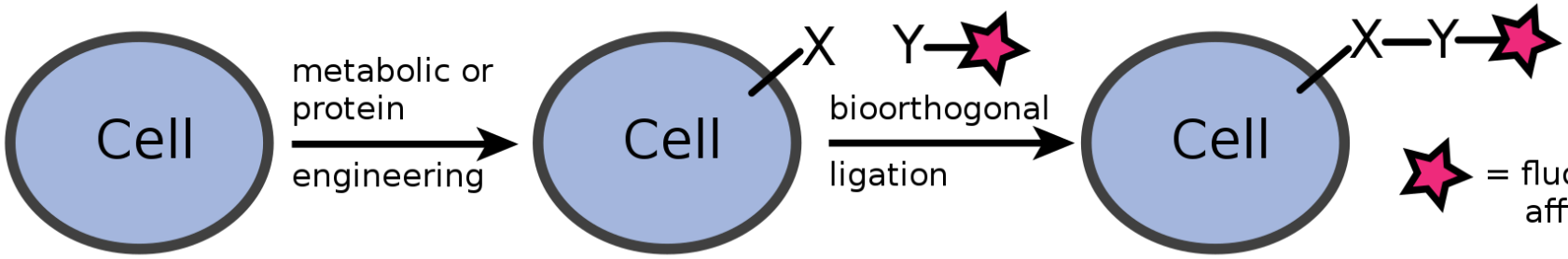




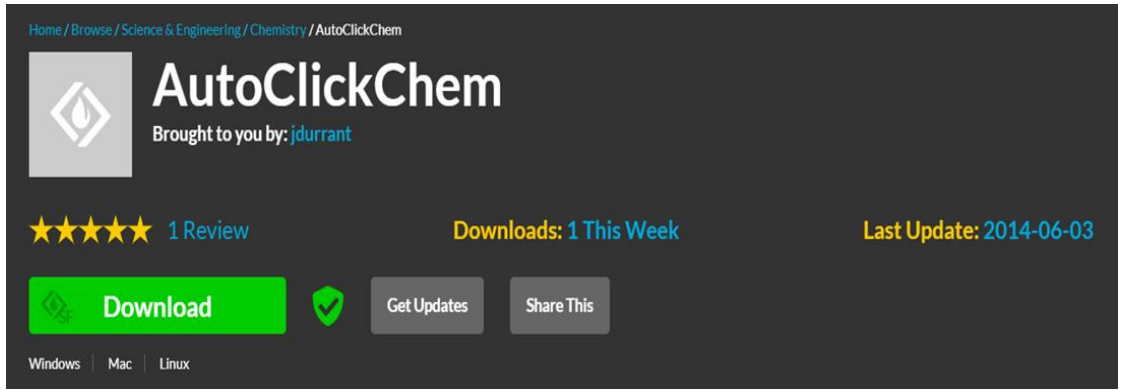


Click reaction and orthogonal chemistry





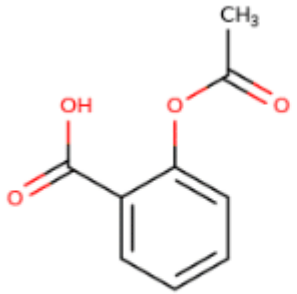
NuroClick software

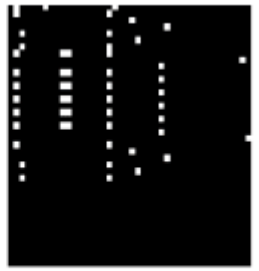


NuroClick software

RDKit → MD ANALYSIS

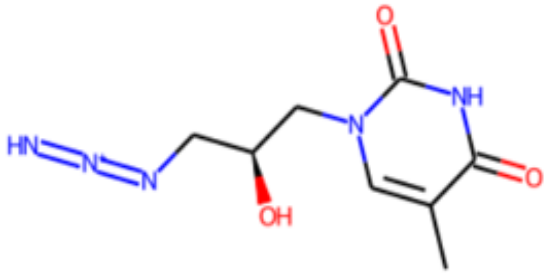
Qt .ui

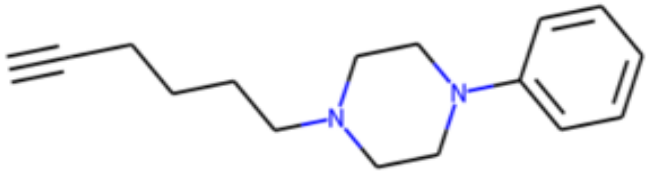


Compound → CC(=O)OC1=CC=CC=C1C(O)=O (SMILES notation) →  (SMILES feature matrix)

How it works?

Step 1: reagents upload





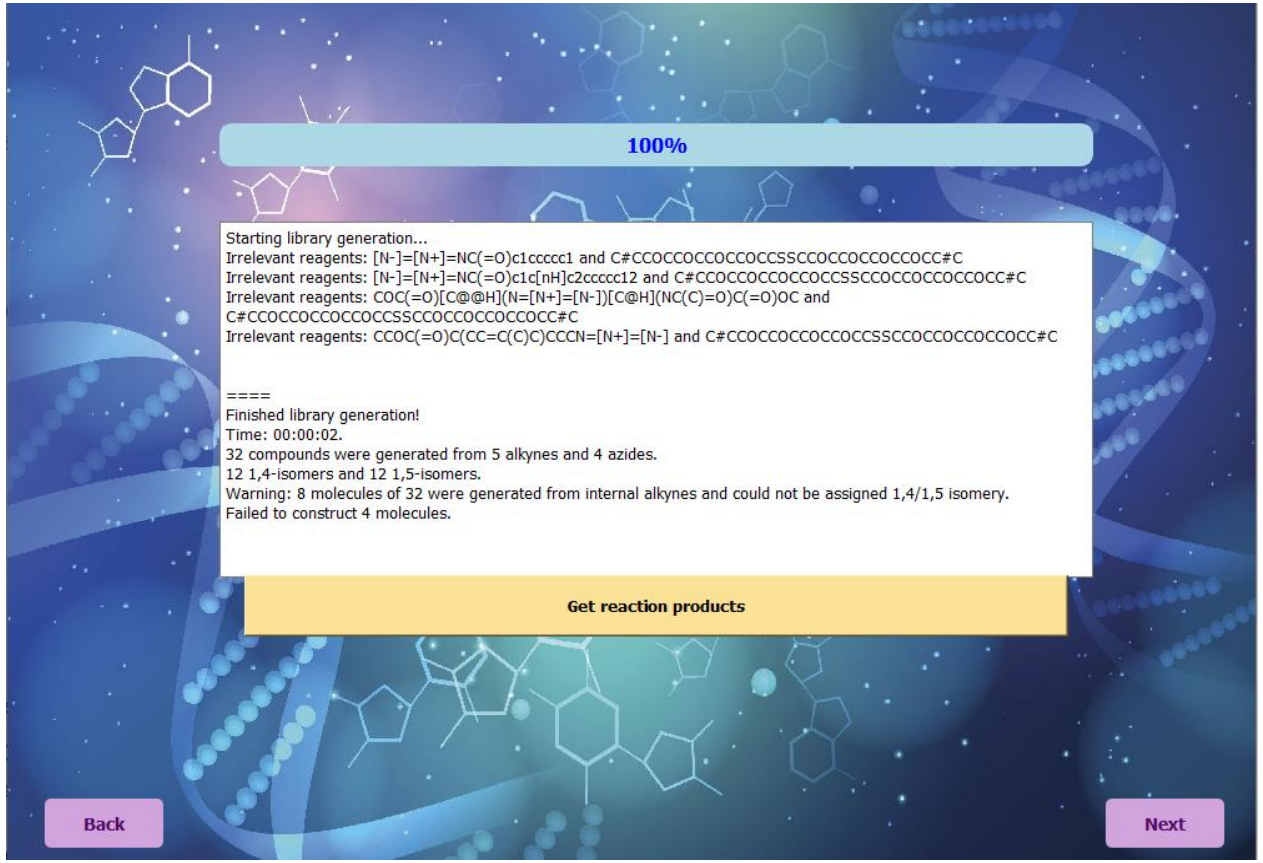
C1=CC=C(C=C1)C(=O)N=[N+]=[N-]
C1=CC=C2C(=C1)C(=CN2)C(=O)N=[N+]=[N-]
CC(=O)N[C@@H]([C@@H]
(C(=O)OC)N=[N+]=[N-])C(=O)OC
CCOC(=O)C(CCCN=[N+]=[N-])CC=C(C)C

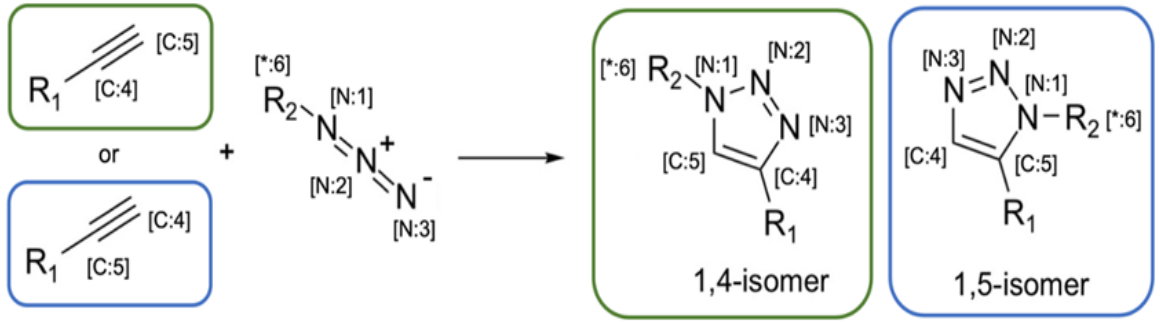
Upload azides

COC(=O)C1=CC=C(C=C1)C#CC2CCNCC2
C#CC(C(=O)NCCOCCOCC(=O)O)N
CC(C)(C)OC(=O)CCOCCC#C
C#CCOC1=CC=C(C=C1)N2C(=O)NNC2=O
C#CCOCCOCCOCCOCCSSCCOCCOCCOCCOCC
#C

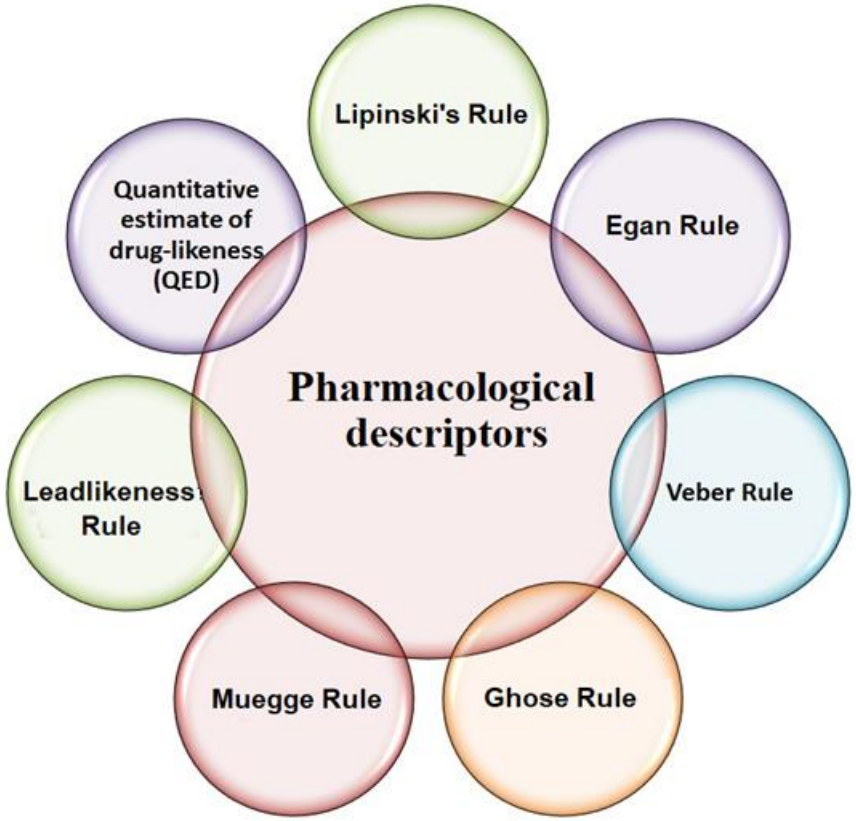
Upload alkynes

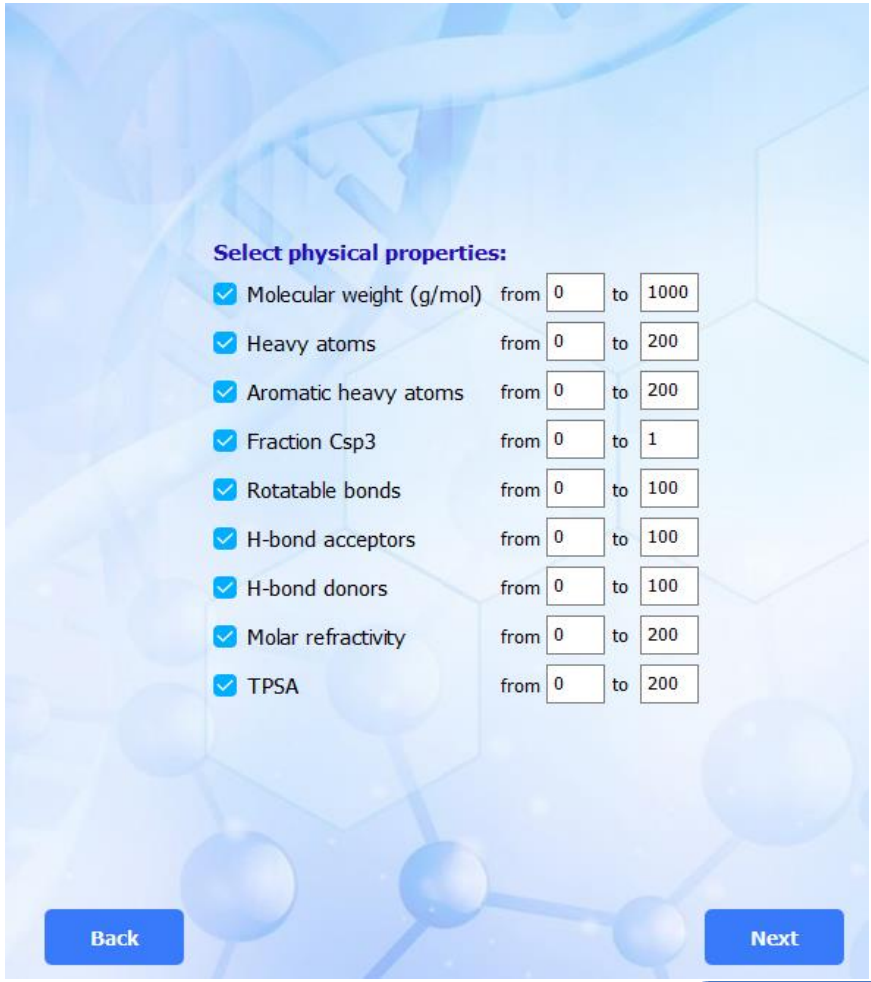
Step 2: click reaction initiation



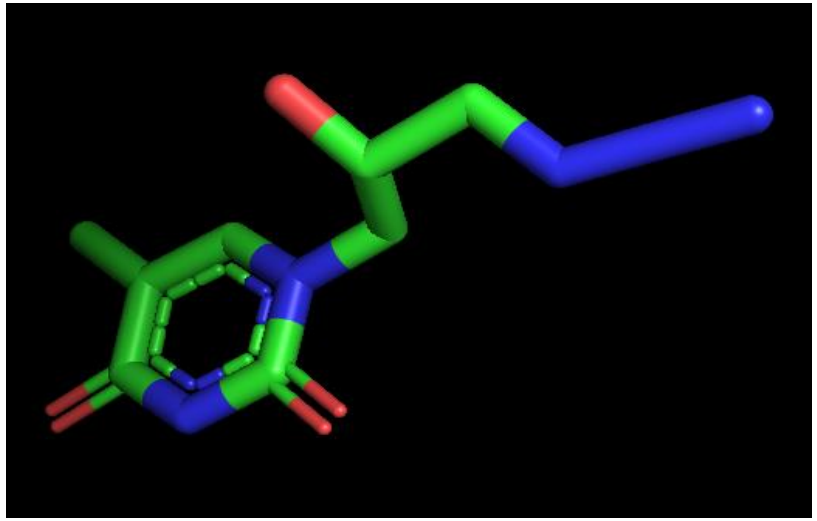


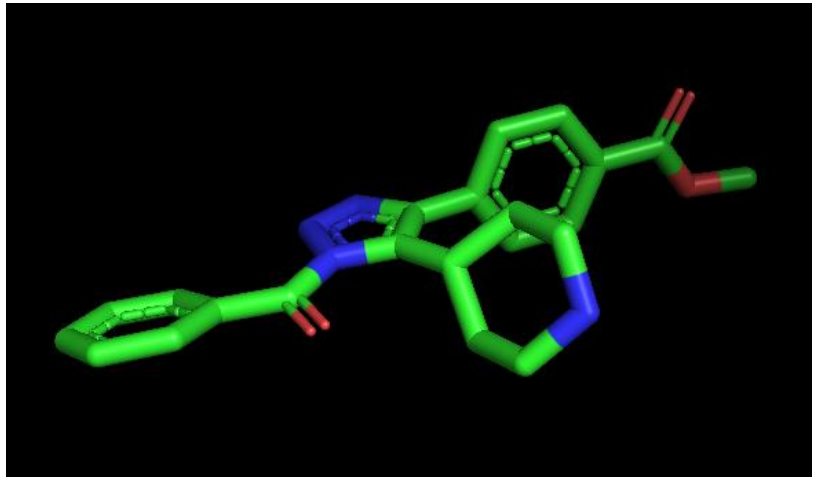
Step 3: PK/PD descriptor calculation

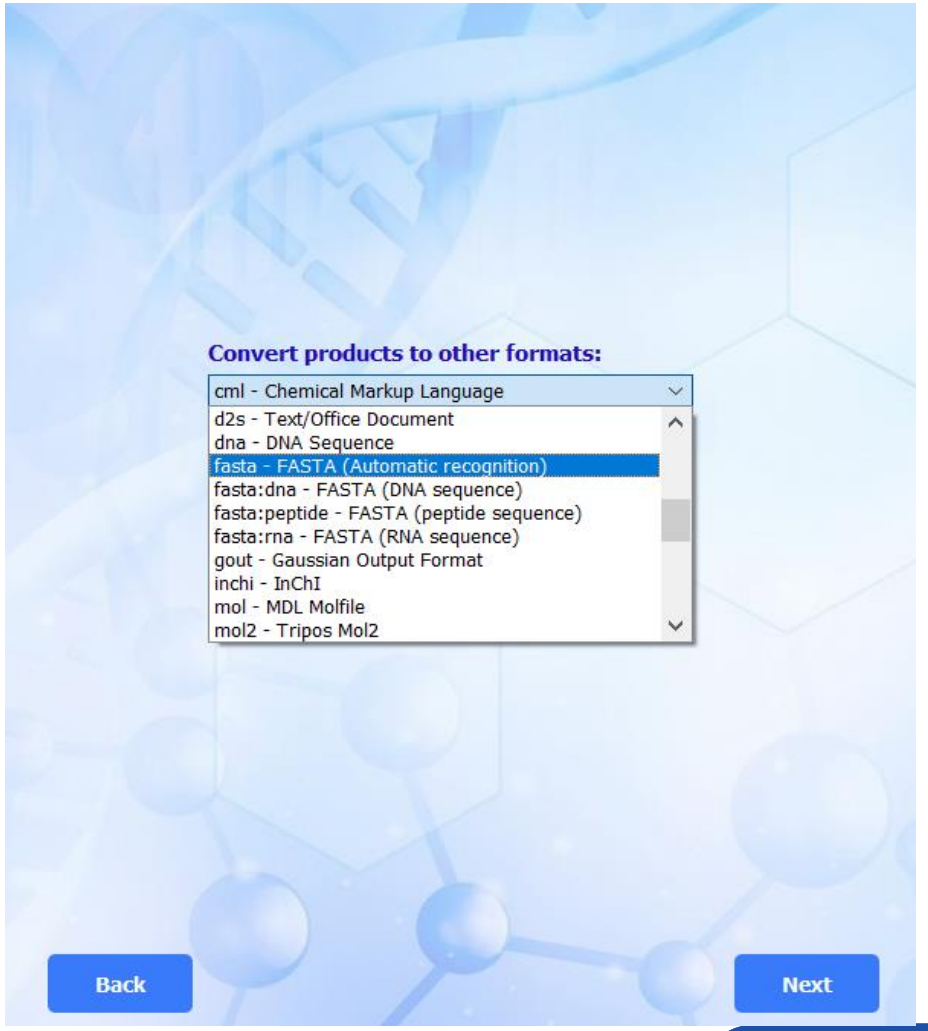




Step 4: save the results







ITMO UNIVERSITY

INFOCHEMISTRY SCIENTIFIC CENTER

Thank you for your attention