

# Phi-CRISPR: Arquitectura Teórica Preliminar

LASU

9 de diciembre de 2025

## Resumen

El presente informe desarrolla el marco conceptual, técnico y bioético del sistema Phi-CRISPR, una arquitectura teórica avanzada concebida como evolución de las tecnologías CRISPR convencionales. Su diseño se orienta a maximizar la precisión, la estabilidad funcional y el control espacio-temporal de la edición genómica, integrando principios de regulación epigenética y modulación génica programable. El documento mantiene un enfoque estrictamente teórico, excluyendo deliberadamente protocolos experimentales o instrucciones técnicas que permitan su reproducción operativa, en consonancia con criterios de responsabilidad científica y normativas bioéticas internacionales.

## 1 Introducción General al Sistema Phi-CRISPR

Phi-CRISPR se plantea como una arquitectura conceptual destinada a ofrecer un control refinado sobre la regulación génica, integrando mecanismos epigenéticos, señales contextuales y moduladores programables. Su objetivo central es lograr intervenciones de alta precisión sin recurrir a mecanismos operativos no regulados. Se presenta como un marco teórico para exploración científica responsable.

## 2 Módulo Terapéutico Conceptual: Eje p53–BRCA1

Este módulo describe una propuesta conceptual destinada a comprender, modelar y justificar estrategias teóricas para potenciar las funciones de TP53 y BRCA1 sin detallar procedimientos experimentales.

### 2.1 Objetivo del módulo

#### 2.1.1. Ficha conceptual: Bifunción p53–BRCA1

El objetivo es diseñar una entidad farmacológica, ya sea una única molécula bifuncional o una combinación controlada, que cumpla dos funciones complementarias. La primera consiste en estabilizar y/o reactivar la actividad transcripcional de p53 (TP53) en células que conserven p53 funcional o variantes susceptibles de reactivación. La segunda es aumentar la expresión funcional o la actividad operativa de BRCA1 mediante la remoción de silenciamiento epigenético del promotor o la estabilización de complejos cofactores que favorezcan su transcripción o estabilidad proteica.

La motivación de este diseño radica en la complementariedad biológica entre ambas proteínas. Mientras p53 actúa como centinela del daño en el ADN y regulador clave de apoptosis y arresto del ciclo ce-

lular, BRCA1 desempeña un papel central en la reparación por recombinación homóloga. Potenciar simultáneamente estas funciones incrementaría la capacidad de reparación y reduciría la supervivencia de clones celulares con inestabilidad genómica.

- Estabilice y/o reactive la actividad transcripcional de p53 (TP53).
- Aumente la expresión o función operativa de BRCA1 mediante regulación epigenética o estabilización proteica.

Motivación: p53 actúa como centinela de daño en ADN; BRCA1 es clave en la recombinación homóloga. Su potenciación conjunta favorece estabilidad genómica.

### 2.2 Estrategias moleculares conceptuales

A. Inhibidor competitivo de p53–MDM2 (tipo Nutlin). B. Reactivador conceptual de p53 mutante (APR-246 / eprenetapopt). C. “Molecular glue” para estabilizar interacciones que favorezcan BRCA1. D. Modulador epigenético dirigido para reducir silenciamiento del promotor BRCA1. E. Diseño híbrido: combinación conceptual de elementos de las estrategias anteriores.

### 2.3 Propiedades físico-químicas objetivo

Rangos guía para diseño in silico:

- MW <550 Da (monofuncional).
- LogP entre 0 y 4.
- PSA <140 Å<sup>2</sup>; ideal <110 Å<sup>2</sup> para acceso nuclear.
- Estado de carga neutro o ligeramente básico.
- Cumplimiento flexible de criterios de drug-likeness.

### 3 Diseño estructural conceptual

Tres fragmentos funcionales componen el esquema bifuncional propuesto. **Dominio A (ancla PPI)**: fragmento mimético del motivo de p53 capaz de ocupar el bolsillo de MDM2 o interactuar con su superficie, reduciendo la degradación de p53. **Enlace/Spacer**: segmento flexible o semirrígido que permita que la otra mitad alcance su blanco sin obstaculizar la unión del Dominio A, con longitud y rigidez optimizadas mediante modelado conformacional. **Dominio B (efector BRCA1)**: dos alternativas conceptuales. B1: superficie farmacófora que actúe como molecular glue reclutando un coactivador transcripcional sobre regiones reguladoras de BRCA1. B2: módulo capaz de favorecer la activación local de cromatina mediante interacción con complejos que disminuyan metilación o aumenten acetilación sobre el promotor de BRCA1.

- B1: superficie tipo molecular glue.
- B2: módulo conceptual de interacción epigenética local.

#### 3.1 Evaluación computacional (conceptual)

Las actividades in silico recomendadas incluyen:

- Docking a MDM2 o variantes de p53.
- Modelado de ternary complex para glues.
- Dinámica molecular y estimación de energía libre.
- ADMET in silico.
- QSAR y modelos ML para priorización.

#### 3.2 Biomarcadores conceptuales

- Aumento de p21 (CDKN1A) como señal de p53 funcional.
- Incremento de BRCA1 a nivel mRNA y proteína.
- Mejora de foci de RAD51.
- Reducción de H2AX en dinámica temporal.

#### 3.3 Riesgos y puntos críticos

- Activación indiscriminada de p53/BRCA1 en tejidos sanos.
- Ineficacia en tumores con delecciones estructurales.
- Posibles off-targets en glues.
- Limitaciones farmacocinéticas por tamaño molecular.

#### 3.4 Alternativas conceptuales

- Terapia combinada conceptual: reactivador p53 + modulador epigenético BRCA1.
- Enfoque de protein rescue para BRCA1.
- Modulación génica no proteica (ARN/epigenética).

### 4 Anexos

#### 4.1 SDRP: Sistema de Detección y Reconocimiento de Patologías

El SDRP es un módulo conceptual de análisis de imágenes para evaluar cambios morfológicos mediante algoritmos de reconocimiento visual.

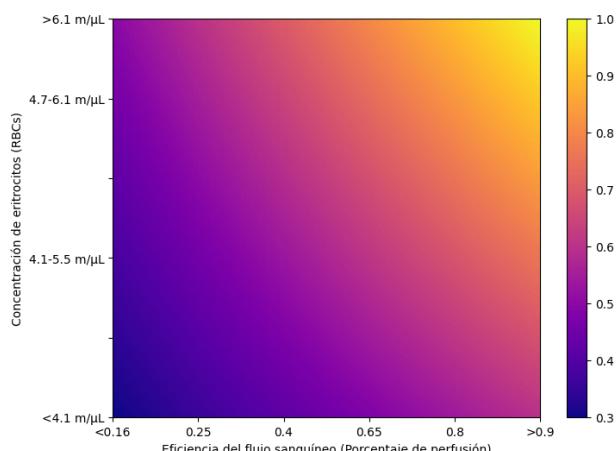


Figura 1: Aproximación genérica por SDRP durante etapa 3 en función de RBCs.

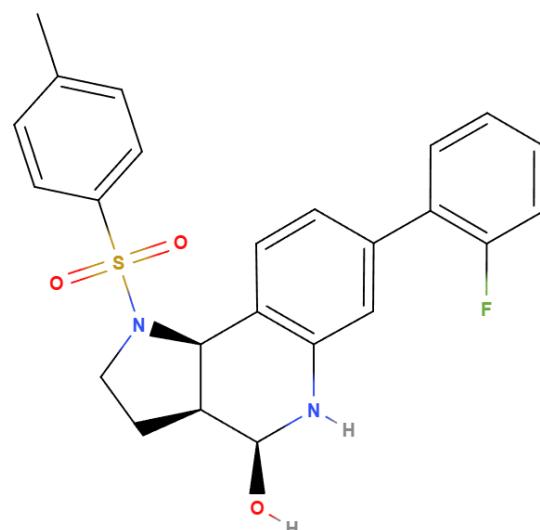


Figura 2: Inhibidor BRD0539 (referencia ilustrativa).

Listing 1: Boceto estructural de SDRP (referencial)

```

1 pip install opencv-python numpy
2     matplotlib
3 import cv2
4 import numpy as np
5 try:
6     image_path1 = 'image1.jpg'
7     image_path2 = 'image2.jpg'
8     img1_original = cv2.imread(
9         image_path1)
10    img2_original = cv2.imread(
11        image_path2)
12    if img1_original is None or
13        img2_original is None:
14        print("Creating dummy images for
15            demonstration as actual
16            images were not found.")
17        dummy_img_shape = (400, 600, 3)
18        img1_original = np.zeros(
19            dummy_img_shape, dtype=np.
20            uint8) + 50
21        img2_original = np.zeros(
22            dummy_img_shape, dtype=np.
23            uint8) + 100
24        img2_original[100:150, 100:200]
25            = [0, 0, 255]
26        cv2.imwrite(image_path1,
27            img1_original)
28        cv2.imwrite(image_path2,
29            img2_original)
30        print(f"Dummy images '{
31            image_path1}' and '{
32            image_path2}' created.")
33        img1_original = cv2.imread(
34            image_path1)
35        img2_original = cv2.imread(
36            image_path2)
37    except Exception as e:
38        print(f"An error occurred while
39            loading/creating images: {e}")
40        print("Creating default dummy images
41            .")
42        dummy_img_shape = (400, 600, 3)
43        img1_original = np.zeros(
44            dummy_img_shape, dtype=np.uint8)
45            + 50
46        img2_original = np.zeros(
47            dummy_img_shape, dtype=np.uint8)
48            + 100
49        img2_original[100:150, 100:200] =
50            [0, 0, 255]
51    if img1_original is None:
52        print(f"Error: Could not load image
53            from '{image_path1}'. Please
54            check the path.")
55    if img2_original is None:
56        print(f"Error: Could not load image
57            from '{image_path2}'. Please
58            check the path.")
59    if img1_original is not None and
60        img2_original is not None:
61        print("Both images loaded
62            successfully.")
63        print(f"Image 1 original shape: {
64            img1_original.shape}")
65        print(f"Image 2 original shape: {
66            img2_original.shape}")
67        img1_gray = cv2.cvtColor(
68            img1_original, cv2.COLOR_BGR2GRAY
69            )
70        img2_gray = cv2.cvtColor(
71            img2_original, cv2.COLOR_BGR2GRAY
72            )
73        print("Images converted to grayscale
74            .")
75        if img1_gray.shape != img2_gray.
76            shape:
77            print("Image dimensions differ.
78                Resizing...")
79        target_height, target_width =
80            img1_gray.shape
81        img2_gray_resized = cv2.resize(
82            img2_gray, (target_width,
83                target_height), interpolation
84                =cv2.INTER_AREA)
85        img1_processed = img1_gray
86        img2_processed =
87            img2_gray_resized
88        print(f"Image 2 resized from {
89            img2_gray.shape} to {
90            img2_processed.shape}.")
91    else:
92        img1_processed = img1_gray
93        img2_processed = img2_gray
94        print("Images already have
95            compatible dimensions.")
96        print("Preprocessed images stored in
97            'img1_processed' and '
98            img2_processed'.")
99    else:
100        print("Cannot proceed with image
101            preprocessing due to loading
102            errors.")
103 from skimage.metrics import
104     structural_similarity as ssim
105 (score, diff) = ssim(img1_processed,
106     img2_processed, full=True)
107 print(f"SSIM Score: {score:.4f}")
108 diff = (diff * 255).astype("uint8")
109 diff_normalized_uint8 = diff
110 print("Difference image calculated and
111            normalized to uint8 as ,
112            diff_normalized_uint8'.")
113 import matplotlib.pyplot as plt
114 fig, axes = plt.subplots(3, 2, figsize
115             =(15, 15))
116 axes[0, 0].imshow(cv2.cvtColor(
117             img1_original, cv2.COLOR_BGR2RGB))
118 axes[0, 0].set_title('Original Image 1')
119 axes[0, 0].axis('off')
120 axes[0, 1].imshow(cv2.cvtColor(
121             img2_original, cv2.COLOR_BGR2RGB))
122 axes[0, 1].set_title('Original Image 2')
123 axes[0, 1].axis('off')
124 axes[1, 0].imshow(img1_processed, cmap='
125             gray')
126 axes[1, 0].set_title('Processed Image 1
127             (Grayscale)')
128 axes[1, 0].axis('off')
129 axes[1, 1].imshow(img2_processed, cmap='
130             gray')
131 axes[1, 1].set_title('Processed Image 2
132             ')
133

```

```
71     (Grayscale)')
72 axes[1, 1].axis('off')
73 axes[2, 0].imshow(diff_normalized_uint8,
74     cmap='gray')
75 axes[2, 0].set_title('Difference Image (
76     SSIM)')
77 axes[2, 0].axis('off')
78 thresh = cv2.threshold(
79     diff_normalized_uint8, 0, 255, cv2.
80     THRESH_BINARY_INV | cv2.THRESH_OTSU)
81 [1]
82 thresh_copy = thresh.copy()
83 contours, _ = cv2.findContours(
84     thresh_copy, cv2.RETR_EXTERNAL, cv2.
85     CHAIN_APPROX_SIMPLE)
86 img1_diff_highlighted = img1_original.
87     copy()
88 for c in contours:
89     if cv2.contourArea(c) > 100:
90         (x, y, w, h) = cv2.boundingRect(
91             c)
92         cv2.rectangle(
```

```
93             img1_diff_highlighted, (x, y)
94             , (x + w, y + h), (0, 0, 255)
95             , 2)
96 axes[2, 1].imshow(cv2.cvtColor(
97     img1_diff_highlighted, cv2.
98     COLOR_BGR2RGB))
99 axes[2, 1].set_title('Image 1 with
100     Highlighted Differences')
101 axes[2, 1].axis('off')
102 plt.tight_layout()
103 plt.show()
```

## 5 Referencias y bibliografía

1. Liu D. Targeting p53–MDM2 interaction by small-molecule inhibitors. J Hematol Oncol. 2008.
2. [Artículo 91] Targeting p53–MDM2 interaction by small-molecule inhibitors: learning from MDM2 inhibitors in clinical trials. J Hematol Oncol. 2022;15:91.