



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS QUÍMICAS

POSGRADO EN CIENCIAS QUÍMICAS
ÁREA: BIOQUÍMICA Y BIOLOGÍA MOLECULAR

Tesis para obtener el grado de:
DOCTOR EN CIENCIAS QUÍMICAS

“Estudio de los determinantes moleculares de la interacción α/β que modulan la cinética del canal de sodio de mamífero”

Presenta:
M. C. Juan Fernando Villa Díaz de León

Directores de Tesis:
DR. Thomas R. F. Scior
DR. Eduardo M. Salinas Stefanón

Julio, 2021

Para la realización de este trabajo se contó con el apoyo del:

Consejo Nacional de Ciencia y Tecnología, CONACyT, para realizar estudios de Doctorado, con numero de CVU: 490356.

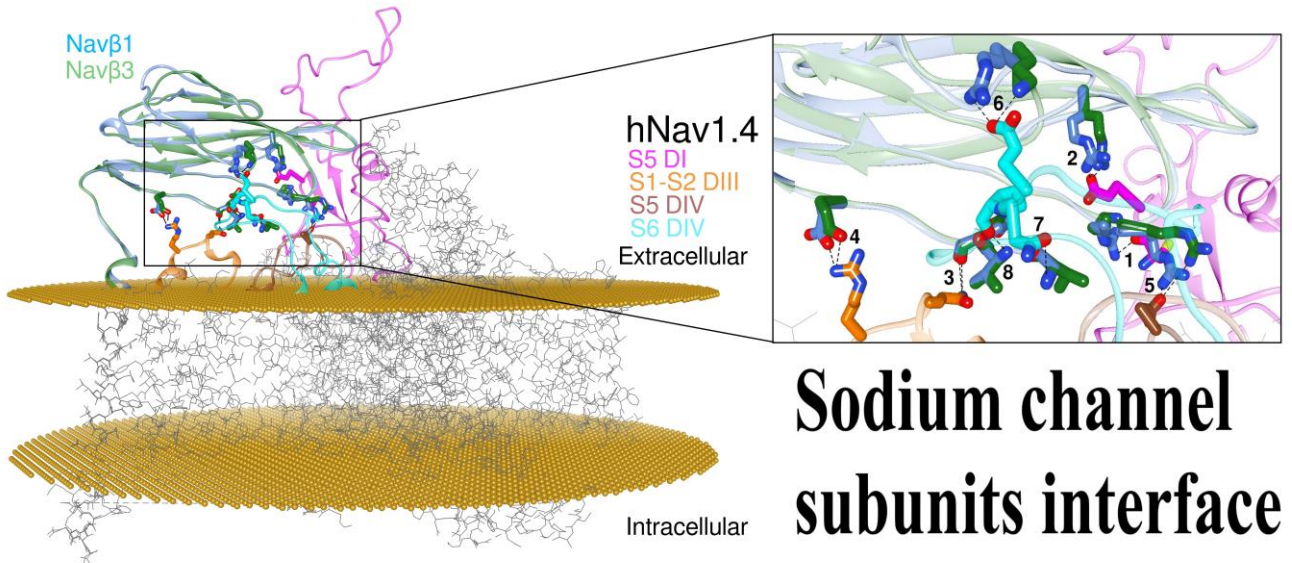
AGRADECIMIENTOS

A mis padres que siempre me han brindado el apoyo incondicional.

A mi familia que siempre me ha brindado el apoyo incondicional.

A Susana López, que me ha apoyado incansablemente.

ABSTRACT



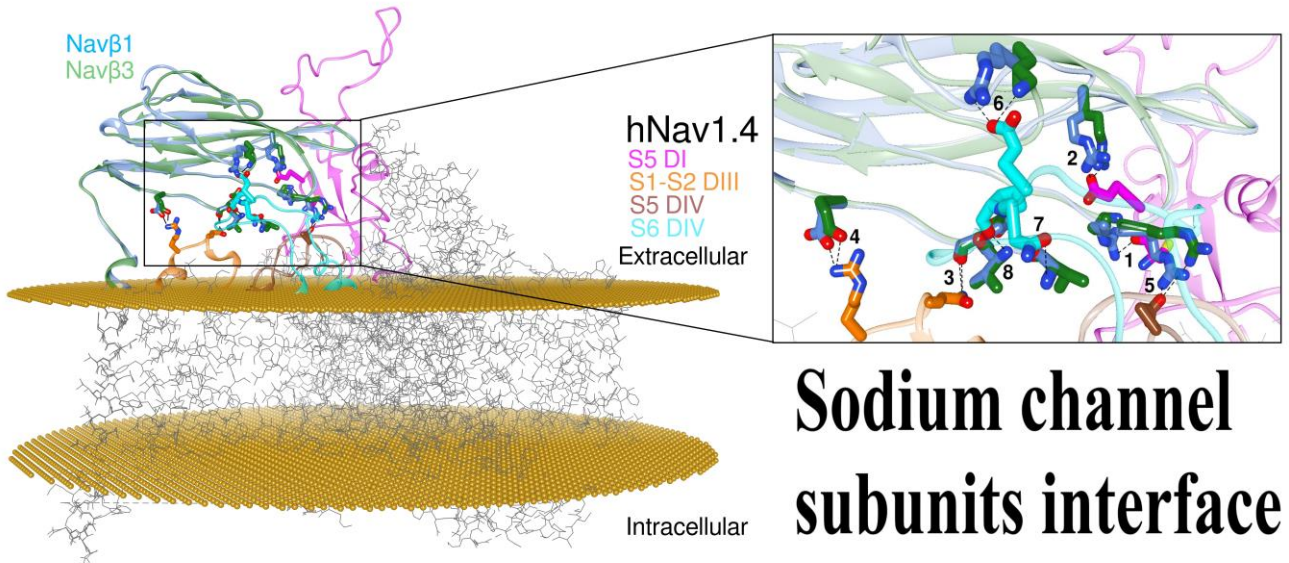
(1) Background: voltage-gated sodium channels (Navs) are integral membrane proteins that allow the sodium ion flux into the excitable cells and initiate the action potential. They comprise an α ($Nav\alpha$) subunit that forms the channel pore and are coupled to one or more auxiliary β ($Nav\beta$) subunits that modulate the gating to a variable extent.

(2) Methods: after performing homology *in silico* modeling for all nine isoforms ($Nav1.1\alpha$ to $Nav1.9\alpha$), the $Nav\alpha$ and $Nav\beta$ protein-protein interaction (PPI) was analyzed chemometrically based on the primary and secondary structures as well as topological or spatial mapping.

(3) Results: our findings reveal a unique isoform-specific correspondence between certain segments of the extracellular loops of the $Nav\alpha$ subunits. Precisely, loop S5 in domain I forms part of the PPI and assists $Nav\beta1$ or $Nav\beta3$ on all nine mammalian isoforms. The implied molecular movements resemble macroscopic springs, all of which explains published voltage sensor effects on sodium channel fast inactivation in gating.

(4) Conclusions: currently, the specific functions exerted by the $Nav\beta1$ or $Nav\beta3$ subunits on the modulation of $Nav\alpha$ gating remain unknown. Our work determined functional interaction in the extracellular domains on theoretical grounds and we propose a schematic model of the gating mechanism of fast channel sodium current inactivation by educated guessing.

RESUMEN



i. Introducción: los canales de sodio dependientes de voltaje (*Navs*) son proteínas integrales que se activan con la despolarización de la membrana celular para iniciar el flujo de iones de sodio en las células excitables e iniciar el potencial de acción.

Los *Navs* son el objetivo molecular de estudio en neurobiología, biología estructural, biología de membranas, biología molecular, electrofisiología, farmacología, proteómica, entre otras áreas. Además, son el blanco molecular de una diversa gama de toxinas y fármacos que se estudian continuamente.

Comprenden de una subunidad α (*Nav α*) que forma el poro del canal y están asociados a una o más subunidades auxiliares β (*Nav β*) que modulan diversos estados del gating.

ii. Métodos: después de realizar el modelado *in silico* por homología para las nueve isoformas (*Nav1.1a* a *Nav1.9a*), la interacción proteína-proteína (PPI) de *Nav α* y *Nav β* se analizó quimométricamente en función de las estructuras primarias y secundarias, así como topológicas o mapeo espacial.

iii. Resultados: los hallazgos revelan una correspondencia específica de isoformas únicas entre ciertos determinantes moleculares de las asas extracelulares de las subunidades *Nav α* . Precisamente, el asa S5 en el dominio I forma parte de PPI y auxilia a *Navβ1* o *Navβ3* en las nueve isoformas de mamíferos. Los movimientos moleculares implícitos se asemejan a resortes macroscópicos, lo cual explica los efectos publicados de los sensores de voltaje sobre la inactivación rápida del canal de sodio en el gating.

iv. Conclusiones: actualmente, se desconocen las funciones mecánicas-estructurales específicas que ejercen las subunidades *Navβ1* o *Navβ3* sobre la modulación del gating de los *Navs*. Nuestro trabajo determinó la interacción funcional en el dominio extracelular sobre bases teóricas y proponemos un modelo esquemático del mecanismo del gating en la inactivación de la corriente de sodio de los Na_v s mediante razonamiento de conjeturas.

ÍNDICE GENERAL

1. Introducción	1
1.1. Canales de sodio dependientes de voltaje	1
1.1.1. La subunidad Nava	3
1.1.2. Las subunidades Nav β	6
1.1.3. Interfaz Nava/Nav β	7
2. Planteamiento del problema	9
3. Justificación	11
4. Antecedentes	12
5. Objetivos	14
5.1. General	14
5.2. Específicos	14
6. Hipótesis	15
7. Material y métodos	16
7.1. Modelado por homología in silico	16
7.2. Identificación de residuos que interactúan en la interfaz entre Nava y Nav β	16
7.3. Modelado in silico de estructuras molde para los modelos de PPI	17
7.4. Determinación de las regiones extracelulares de las Navas	17
7.5. Cálculo de las propiedades de las ECLs	17
7.6. Cálculo de las propiedades químicas en la superficie de las IF-ECL	18
7.7. Interacciones electrostáticas de las superficies IF-ECL (MEPS)	18
7.8. Cálculo de propiedades de interfaz de Nava/Nav β 1 y Nava/Nav β 3	19
7.9. Validación del modelo externo de los modelos PPI	19
8. Resultados	20
8.1. Determinación de las PPI en las isoformas de los Navs	20
8.1.1. Análisis PPI del complejo estructural eeNav1.4 α /eeNav β 1	22
8.1.2. Análisis PPI de los modelos hNav1.4 α /hNav β 1 y hNav1.4 α /hNav β 3	22
8.1.3. Identificación de los residuos en las Navas	24
8.1.4. Alineación estructural de las Nav β s	26
8.2. Modulación de la aceleración de la inactivación rápida del gating del poro por Nav β	29
8.3. Determinación de las propiedades de las ECLs	34
8.4. Propiedades de interfaz y Patrones de interacción en los Navs	42
9. Discusión de resultados	48
9.1. Patrones de PPI en las isoformas de los Navs	48
9.2. Aceleración hipotética de la inactivación rápida del gating de los Navs	55
9.3. Propiedades de los residuos de las ECLs en las isoformas Navas	57
9.4. Propiedades de volumen y superficie de las ECLs en las isoformas Navas	58
9.5. Propiedades de interfaz de Nava/Nav β 1 y Nava/Nav β 3	59
10. Conclusiones	62
11. Referencias	65
12. Material suplementario	71

ÍNDICE DE TABLAS

<i>Tabla 1. Enfermedades y distribución en el tejido de las subunidades Nava.</i>	2
<i>Tabla 2. Enfermedades y distribución en el tejido de las subunidades Navβ.</i>	6
<i>Tabla 3. Sinopsis de los patrones de PPI estudiados.</i>	21
<i>Tabla 4. Alineamiento de secuencia múltiple de los PPI-Id.</i>	25
<i>Tabla 5. Residuos de contacto del S4 DIII con las subunidades Navβ1 y Navβ3.</i>	32

ÍNDICE DE FIGURAS

<i>Figura 1. Inhibición electrofisiológica de la apertura del canal Nav por TTX.</i>	3
<i>Figura 2. Topología y estructura de Nava.</i>	4
<i>Figura 3. Representación esquemática del gating.</i>	5
<i>Figura 4. Estructura de Navβ1 en complejo con hNavα1.4.</i>	7
<i>Figura 5. Visualización del modelo 3D con vista desde el ECL S6 DIV de eeNav1.4α/eeNavβ1.</i>	22
<i>Figura 6. Visualización de los modelos 3D con vista desde el ECL S6 DIV de hNav1.4α/hNavβ1.</i>	23
<i>Figura 7. Alineamiento estructural de los ectodominios (IgD) de las estructuras molde de Navβs.</i>	28
<i>Figura 8. Sensor de voltaje S4 DIII de hNav1.4α en estrecho contacto con hNavβ1 o hNavβ3.</i>	31
<i>Figura 9. Modulación hipotética de la inactivación rápida del gating Nava por Navβ1 o Navβ3.</i>	34
<i>Figura 10. Longitud de las ECLs de las Navas.</i>	34
<i>Figura 11. Propiedades de los residuos de la ECR de las Navas.</i>	35
<i>Figura 12. Propiedades de los residuos de las ECLs de las Navas.</i>	36
<i>Figura 13. Propiedades de SAA y volumen de la ECR y ECLs S5 y S6 de las Navas.</i>	37
<i>Figura 14. SAA de las ECLs de las Navas.</i>	38
<i>Figura 15. Volumen molecular de las ECLs de las Navas.</i>	39
<i>Figura 16. Propiedades de SAA de la ECR y las IF-ECL de las Navas.</i>	41
<i>Figura 17. Área total de los átomos que forman la interfaz Nava/Navβ.</i>	42
<i>Figura 18. Porcentajes de las propiedades de superficie de interfaz de Navβ1 y Navβ3 en complejo con el S5 DI.</i>	43
<i>Figura 19. Porcentajes de las propiedades de superficie de interfaz de Navβ1 y Navβ3 en complejo con el S1-S2 DIII.</i>	44
<i>Figura 20. Porcentajes de las propiedades de superficie de interfaz de Navβ1 y Navβ3 en complejo con el S5 DIV.</i>	45
<i>Figura 21. Porcentajes de las propiedades de superficie de interfaz de Navβ1 y Navβ3 en complejo con el S6 DIV.</i>	46
<i>Figura 22. Ejemplo demostrativo de las propiedades de superficie de los átomos que forman la interfaz en hNavβ1/S5 DI del complejo hNavβ1/hNav1.4α.</i>	47
<i>Figura 23. Presentación de un modelo de hNavβ1 trimérico.</i>	50
<i>Figura 24. Superposición de proteínas Navβ monoméricas y triméricas en complejo con Nava.</i>	51
<i>Figura 25. Ubicación tridimensional (3D) de las subunidades Navβ1 y Navβ3 en la subunidad formadora del poro Nava.</i>	52

ABREVIATURAS

Navs	<i>Canales de sodio dependientes de voltaje</i>
Nava	<i>Subunidad alfa de los canales de sodio dependientes de voltaje</i>
Navβ	<i>Subunidad beta de los canales de sodio dependientes de voltaje</i>
TTX	<i>Tetradotoxina</i>
DI a DIV	<i>Dominios 1 a 4</i>
S1 a S6	<i>Segmentos transmembrana helicoidales 1 a 6</i>
VSD	<i>Dominio de detección de voltaje</i>
S4	<i>Sensor de voltaje</i>
ECR	<i>Región extracelular</i>
ECL	<i>Asa extracelular</i>
IF-ECL	<i>Asa extracelular que participa en la interfaz con Navβ</i>
IF	<i>Interfaz</i>
IgD	<i>Dominio de inmunoglobulina</i>
TMH	<i>Hélice transmembrana</i>
cryo-EM	<i>Microscopía crio-electrónica</i>
/	<i>El símbolo “/” esta designado para indicar una interfaz entre dos moléculas ó átomos</i>
CRX	<i>Cristalografía de rayos X</i>
PPI	<i>Interacción proteína-proteína</i>
PPI-Id	<i>Números designados a específicos PPIs entre residuos</i>
MEPS	<i>Potencial electrostatico de superficie molecular</i>
P-MEPS	<i>MEPS positivo</i>
P-MEPS	<i>MEPS positivo</i>
SAA	<i>Área accesible al solvente</i>

1. INTRODUCCIÓN

“Un modelo debe estar mal, en algunos aspectos, de lo contrario sería la misma cosa en sí. El truco consiste en ver dónde está bien”.

Henry A. Bent

1.1. Canales de sodio dependientes de voltaje

Concerniente a los canales de sodio dependientes de voltaje (Navs), se ha informado una gran cantidad de genes, sus marcos de lectura, patrones de expresión y funciones para una gran diversidad de organismos que van desde células procariotas a eucariotas. Tratamos a las isoformas de los canales iónicos en vertebrados con sus mayores complejidades estructurales y génicas [Catterall *et al.*, 2005; Fux *et al.*, 2018].

Los organismos multicelulares complejos requieren una transmisión rápida y precisa de información entre células y tejidos, y una estrecha coordinación de funciones distantes [Yu y Catterall, 2004]. Los Navs son responsables de incrementar la permeabilidad de Na⁺, que consecuentemente inicia el potencial de acción que media la transmisión rápida de impulsos eléctricos en células eléctricamente excitables de tejidos (e. g. nervio, corazón y musculo esquelético) y son el blanco molecular de una amplia gama de toxinas y fármacos [Catterall, 1980; Cestèle y Catterall, 2000]. Las corrientes de sodio dependientes de voltaje fueron descubiertas por Hodgkin y Huxley en 1952 cuando estudiaban la conductancia eléctrica en axones de calamares gigantes [Hodgkin y Huxley, 1952]. Veintiséis años después, los Navs fueron aislados y purificados del electroplax de la anguila eléctrica [Agnew *et al.*, 1978].

El complejo de los Nav generalmente consta de una subunidad α central (Nav α) que forma el poro del canal, el cual está codificado por los genes SCN1A a SCN5A (Nav1.1 α a Nav1.5 α , respectivamente) y SCN8A a SCN11A (Nav1.6 α a Nav1.9 α , respectivamente).

Las nueve isoformas de la subunidad Nav α se expresan en patrones de tejido específicos y exhiben diferencias en el comportamiento del *gating* que se adaptan a diferentes funciones fisiológicas [Waxman, 2012; Goldin *et al.*, 2000; O'Malley *et al.*, 2015]. Nav1.1 α a Nav1.3 α , se expresan en el sistema nervioso central. Nav1.6 α , se expresa en los sistemas nerviosos periférico y central; Nav1.7 α a Nav1.9 α , se expresan en el sistema nervioso periférico. Nav1.4 α y Nav1.5 α son abundantes en músculo esquelético y cardiaco, respectivamente (Tabla 1) [Catterall *et al.*, 2005; Cestèle y Catterall, 2000; Goldin, 2000].

Tabla 1. Enfermedades y distribución en el tejido de las subunidades Nava.
Tomada y modificada de Isom y Kruger, 2016

Gen	Isoforma	Distribución en el tejido	Enfermedades causadas por mutaciones
SCN1A	Na _v 1.1 α	CNS	DS, AM, FHM3, FS, GEFS SUDEP
SCN2A	Na _v 1.2 α	CNS	CFBNI, DS, EOEE, AM, GEFS, OS
SCN3A	Na _v 1.3 α	CNS	Unclear
SCN4A	Na _v 1.4 α	ME	PAM, PMC, Hiper PP, Hipo PP, SNEL
SCN5A	Na _v 1.5 α	MC	AF, AP, BS, DCM, LQTS, PCCD, SIDS, SSS, SUDEP
SCN8A	Na _v 1.6 α	CNS, PNS	EOEE; DC, parálisis, ataxia, distonía
SCN9A	Na _v 1.7 α	PNS	CIP, IEM, PEPD, PPN
SCN10A	Na _v 1.8 α	PNS	PPN
SCN11A	Na _v 1.9 α	PNS	PPN

AM: Autismo familiar; ME: Músculo esquelético; MC: Músculo cardiaco; AF: fibrilación auricular; AS: paro auricular; CFBNI: convulsiones familiares benignas neonatales e infantiles; BS: síndrome de Brugada; CIP: insensibilidad asociada a la canalopatía del dolor; CNS: sistema nervioso central; DCM: miocardiopatía dilatada; DC: Deterioro cognitivo; DRG: ganglio de la raíz dorsal; DS: síndrome de Dravet; EOEE: encefalopatía epiléptica de inicio precoz; FHM3: migraña hemipléjica familiar tipo 3; FS: convulsiones febriles; GEFS: epilepsia genética con convulsiones febriles; HyperPP: parálisis periódica hipercalémica, HipoPP: parálisis periódica hipocalémica; IEM: eritromelalgia hereditaria; LQTS: síndrome de QT largo; OS: síndrome de Ohtahara; PAM: miotonía agravada por potasio; PCCD: enfermedad progresiva de la conducción cardíaca; PEPD: trastorno de dolor extremo paroxístico conocido como síndrome de dolor rectal familiar; PMC: paramiotonía congénita; PNS: sistema nervioso periférico; PPN: neuropatías periféricas dolorosas; SIDS: síndrome de muerte súbita infantil; SNEL: laringoespasma episódico neonatal grave; SSS: síndrome del seno enfermo; SUDEP: muerte repentina inexplicada en la epilepsia.

La clasificación farmacológica de las Navas diverge según su sensibilidad y resistencia a la tetrodotoxina (TTX); Nav1.1 α a Nav1.4 α , Nav1.6 α y Nav1.7 α son sensibles al bloqueo por concentraciones nanomolares bajas de TTX (TTX-S); Nav1.5 α , Nav1.8 α y Nav1.9 α son resistentes a concentraciones >1 μ M TTX (TTX-R) [Ogata y Onishi, 2002]. En la Figura 1, se muestra la inhibición de la corriente de Na⁺, en Nav α por TTX.

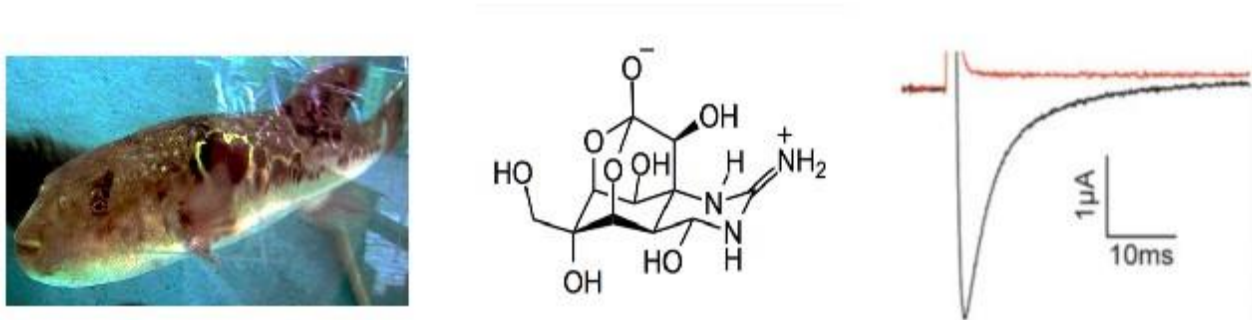


Figura 1. Inhibición electrofisiológica de la apertura del canal Nav por TTX. TTX se encuentra en el pez globo japonés. A la izquierda, *Takifugu rupripes* en un acuario; En el centro, complejo molecular TTX; A la derecha, inhibición de la corriente de rNav1.2 α con un potencial de sostenimiento de -90 a -20 mV expresada en ovocitos de *Xenopus laevis*; En negro se muestra el control, en rojo se muestra la corriente después de adicionarle 10 nM de TTX. Tomada y modificada de Gilchrist *et al.*, 2014.

1.1.1. La subunidad Nava

La subunidad Nav α de los vertebrados consta de una sola cadena polipeptídica con una masa molecular de ~260 kDa. Abarca el componente selectivo de iones que se pliega en cuatro dominios homólogos, pero no idénticos (DI a DIV), cada dominio contiene seis segmentos helicoidales transmembrana (S1 a S6), que se ensamblan alrededor del poro selectivo de iones [Mantegazza y Catterall, 2012; Ahern *et al.*, 2015], Figura 2.

Los segmentos helicoidales transmembrana S1, S2, S3 y S4 comprenden los cuatro dominios de detección de voltaje (VSD) en DI a DIV. Están ubicados en el borde exterior en cada esquina de la subunidad Nav α . La hélice S4 constituye el sensor de voltaje de cada VSD, la cual ha evolucionado hacia un dominio anfipático con una cara cargada positivamente.

En respuesta a los cambios en el campo eléctrico producidos por la despolarización de la membrana, el S4 se mueve hacia la zona extracelular iniciando cambios conformacionales que a su vez abren el poro [Payandeh *et al.*, 2011; Long *et al.*, 2005]. Los S4 (DI-DIII) muestran una cinética más rápida en respuesta a la despolarización que a su vez realizan un

cambio conformacional en el canal que abre el poro y consecuentemente permite que los cationes Na^+ ingresen a las células.

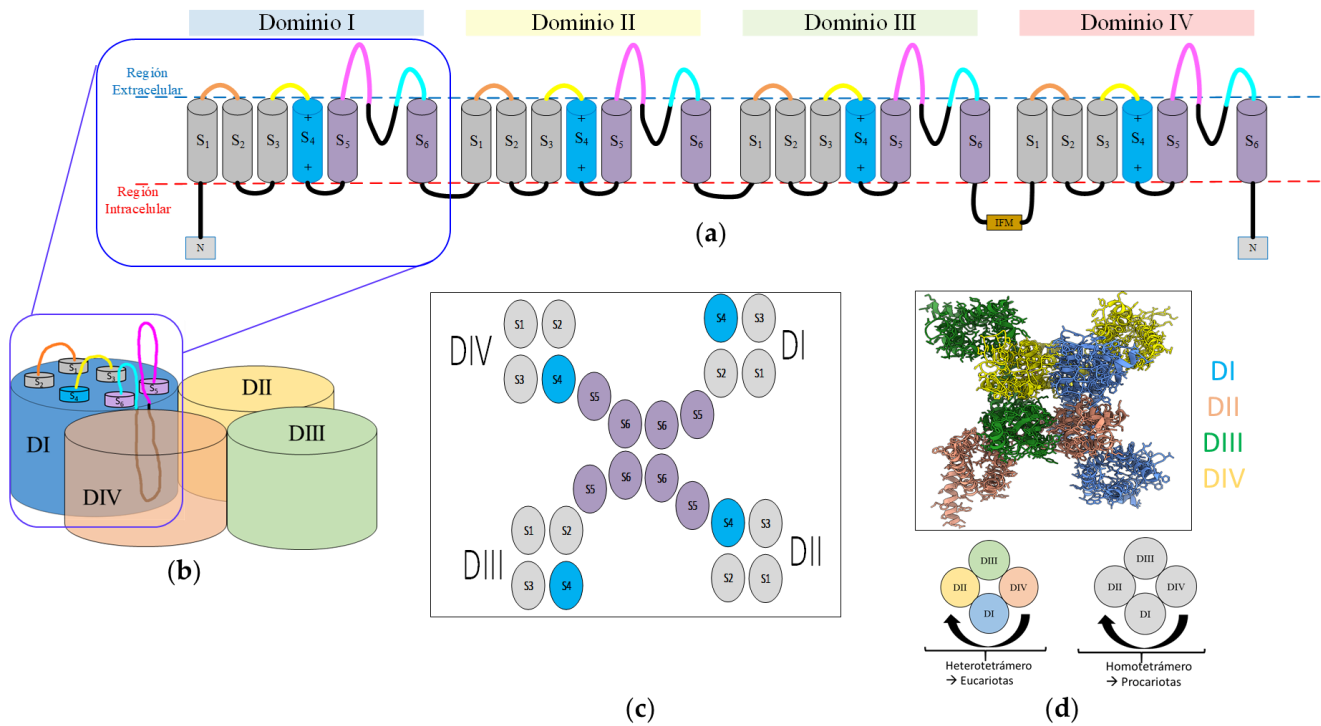


Figura 2. Topología y estructura de *Nav*.

(a) Se muestra la topología de *Nav*; (b) Topología que indica cómo se encuentran ensamblados los 6S transmembrana en cada dominio; (c) Vista por encima de la región extracelular el ensamblaje de los 6S; (d) Estructura resuelta por cryo-EM de de *Nav* (PDB: 6AGF, Pan *et al.*, 2018), nótese que los DI–DIV se encuentran ensamblados en sentido a las manecillas del reloj; Sensor de voltaje en azul cielo; S5 y S6 en morado; Asas extracelulares: S1-S2 en naranja, S3-S4 en amarillo, S5 en magenta, S6 en cian.

El S4 DIV responde más lentamente; su movimiento libera un conector intracelular llamado puerta de inactivación IFM el cual contiene por sus siglas tres residuos lipofílicos (I: isoleucina, F: fenilalanina, M: metionina) y conecta el S6 DIII con el S1 DIV [Chanda y Benzanilla, 2002; Bosmans *et al.*, 2008; Capes *et al.*, 2013]. Como resultado, la puerta de inactivación se mueve para ocluir el poro y lleva al canal a un estado inactivo. Por lo tanto, la activación e inactivación del canal están vinculadas de forma estructural, mecánica y funcional, Figura 3 [Catterall, 2013; Vargas *et al.*, 2012].

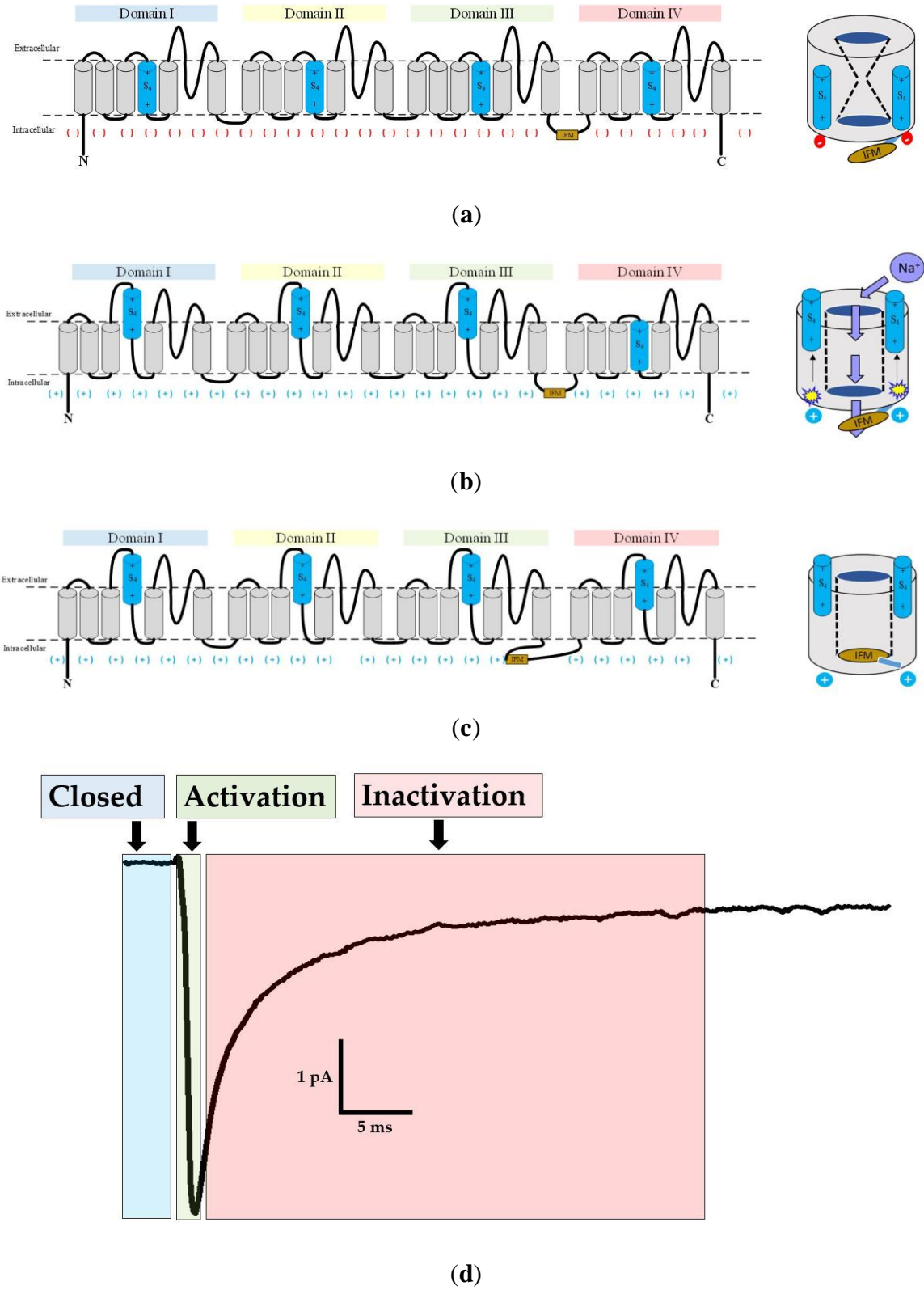


Figura 3. Representación esquemática del gating.

Los tres esquemas de Navs eucariotas muestran los estados (a) cerrado; (b) abierto; y (c) inactivado. Una corriente de membrana típica de *Rattus norvegicus* de la isoforma rNav1.4a responde a un pulso despolarizante que refleja los tres estados principales de activación; Puerta de inactivación de IFM: café oscuro; Sensores de voltaje S4: azul cielo.

1.1.2. Las subunidades Nav β

La mayoría de los Nav de las células de los vertebrados forman unidades biológicas con subunidades β (Nav β s) asociadas. Hay cuatro genes de Nav β s (*SCN1B* a *SCN4B*) que codifican cuatro proteínas Nav β 1 a Nav β 4, respectivamente [Brackenbury e Isom, 2011; Cusdin et al., 2008]. Al igual que las subunidades Nav α , las subunidades Nav β se expresan individualmente en diferentes tejidos, Tabla 2 [Patino e Isom, 2010; Calhoun e Isom, 2014].

Tabla 2. Enfermedades y distribución en el tejido de las subunidades Nav β .
Tomada de Brackenbury e Isom, 2008.

Proteína	Símbolo del gen	Localización en el tejido	Enfermedad relacionada
β 1	<i>SCN1B</i>	CNS, corazón, PNS, músculo esquelético	Epilepsia, arritmia cardíaca, cáncer
β 2	<i>SCN2B</i>	CNS, corazón, PNS	Respuesta alterada del dolor, susceptibilidad a convulsiones, MS
β 3	<i>SCN3B</i>	Glándula suprarrenal, CNS, riñón, PNS	Cáncer
β 4	<i>SCN4B</i>	CNS, corazón, PNS, músculo esquelético	Enfermedad de Huntington

CNS: Sistema nervioso central; MS: Esclerosis múltiple; PNS: Sistema nervioso periférico.

Si bien, la subunidad Nav α es suficiente para la detección de voltaje y la conductancia iónica selectiva, las subunidades Nav β s modulan los valores máximos de la corriente de sodio y modifican la cinética de activación e inactivación de la subunidad Nav α . Aumentan la densidad de corriente máxima mediante el aumento de la densidad de canales abiertos (número por área) en la superficie celular. Cambian efectivamente el rango de voltaje involucrado en la activación e inactivación y mejoran las tasas de inactivación y recuperación de la inactivación [O'Malley e Isom, 2015; Brackenbury e Isom, 2011; Catterall *et al.*, 1995; Isom *et al.*, 1992, 1995; McCormick *et al.*, 1998; Makita *et al.*, 1996; Patton *et al.*, 1994; Yu *et al.*, 2005; Chen *et al.*, 1995].

Anterior a este estudio quimiométrico, se han llevado a cabo experimentos electrofisiológicos y de mutagénesis sitio dirigida en combinación con estudios *in silico* para estudiar la modulación de la subunidad Nav α por Nav β 1 [Islas *et al.*, 2013; Scior *et al.*, 2015; Sánchez-Solano *et al.*, 2016].

Todas las subunidades Nav β son proteínas de membrana de tipo 1. La región amino-terminal extracelular contiene un dominio de inmunoglobulina amino-terminal (IgD) de tipo V único y un cuello corto conectado a una hélice transmembranal (TMH), la cual termina en una región intracelular carboxilo-terminal. Las similitudes de secuencias entre Nav β 1 y Nav β 3 son mayores que entre Nav β 2 y Nav β 4 [Chopra *et al.*, 2007; Isom *et al.*, 2001]. En la Figura 4, se muestra la emulación de la estructura de la subunidad Nav β 1 ensamblada a la membrana celular en complejo con Nav α .

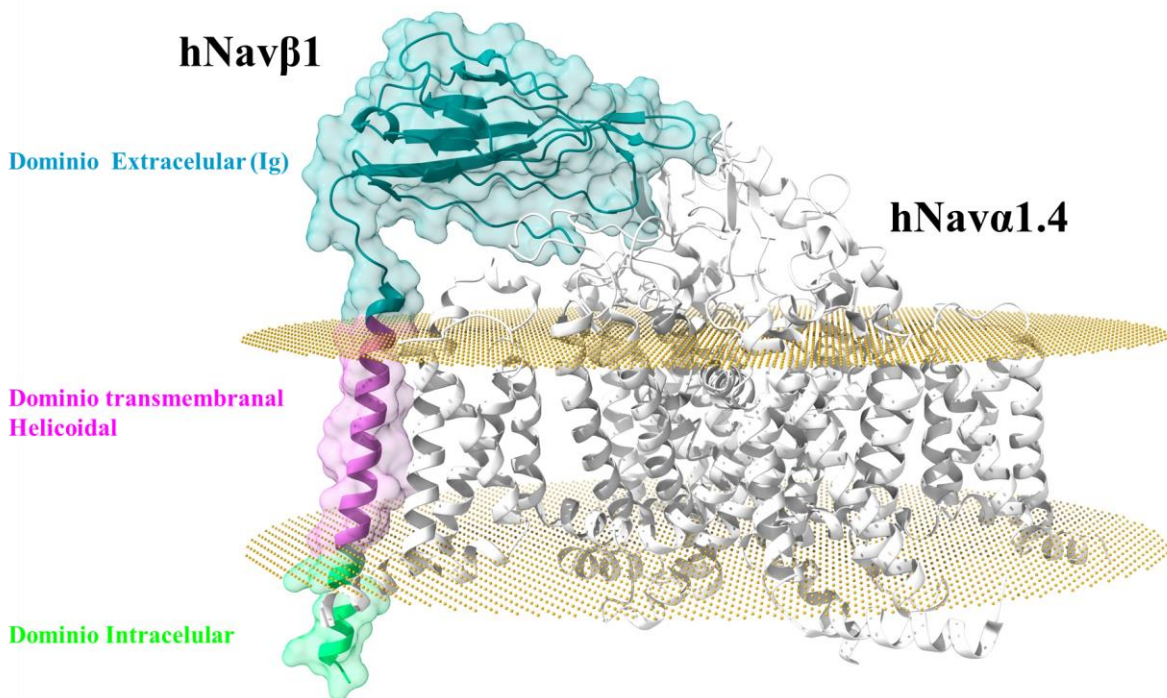


Figura 4. Estructura de Nav β 1 en complejo con hNav α 1.4.

Se muestra la estructura determinada por microscopía crio-electrónica (cryo-EM) de Nav β 1 ensamblada a la membrana celular (representada por puntos en color oro oscuro), en complejo con hNav α 1.4, en blanco (Pan *et al.*, 2018). En transparente se muestra la superficie excluida al disolvente de Nav β 1.

1.1.3. Interfaz Nava/Nav β

Las estructuras determinadas por cryo-EM de los Navs de insecto, anguila eléctrica, rata y humano revelan una arquitectura tridimensional (3D) idéntica de sus subunidades Nava. La comparación de la interfaz entre Nava y Nav β (Nava/Nav β) de anguila eléctrica y de *Homo sapiens* revela que está sorprendentemente bien conservada [Shen *et al.*, 2017; Yan *et al.*, 2017; Pan *et al.*, 2018; Shen *et al.*, 2019; Jiang *et al.*, 2020].

Las subunidades Nav β 1 y Nav β 3 están unidas a Nav α a través de interacciones no covalentes, mientras que Nav β 2 y Nav β 4 están vinculados covalentemente por uno o más puentes de disulfuro con Nav α [Buffington y Rasband *et al.*, 2013; Chen *et al.*, 2012].

La identificación de los sitios de interacción para la modulación de la subunidad Nav α con las subunidades Nav β se deriva principalmente del análisis mutagénico y la información estructural de las subunidades Nav β individuales [Makita *et al.*, 1996; Islas *et al.*, 2013; Scior *et al.*, 2015; Sánchez-Solano *et al.*, 2016; Chen *et al.*, 2012; Pan *et al.*, 2018; Yu *et al.*, 2003; Gilchrist *et al.*, 2013; Namadurai *et al.*, 2013; Das *et al.*, 2016; McCormick *et al.*, 1999; Yereddi *et al.*, 2012].

2. PLANTEAMIENTO DEL PROBLEMA

Actualmente, sigue siendo una pregunta sin respuesta: ¿Cómo exactamente la subunidad formadora del poro ($Nav\alpha$) es modulada por las subunidades $Nav\beta 1$ y $Nav\beta 3$ en un sentido estricto mecánico-estructural? El presente estudio quimiométrico tiene como objetivo encontrar patrones de interacción para describir el comportamiento mecanicista en $Nav\alpha/Nav\beta$ que podría brindar un nuevo panorama sobre la modulación de las nueve isoformas ($Nav1.1\alpha$ a $Nav1.9\alpha$) comparando tres especies de mamíferos; *Homo sapiens* (hNav), *Mus musculus* (mNav) y *Rattus norvegicus* (rNav). Para ello, se aplicaron métodos *in silico* bien establecidos, como alineamientos de secuencias múltiples (MSA), determinaciones estructurales, análisis de propiedades de aminoácidos, generación de modelos por homología y potenciales electrostáticos de superficie molecular (MEPS), que pueden codificarse por colores y proyectarse sobre las superficies moleculares.

Para adaptarse a nuestro análisis quimiométrico, la subunidad α formadora del poro se clasifica en las siguientes partes topológicas; la subunidad α formadora del poro se denomina ($Nav\alpha$ o α para en su forma abreviada. A su vez, se compone de tres segmentos topológicos: (i) la parte o región extracelular (ECR) con sus 16 asas extracelulares (ECLs); (ii) la parte TMH (hélices transmembrana); y finalmente (iii) la región intracelular.

Sobre una base teórica, determinamos las características estructurales (3D) y los patrones de secuencia, así como las propiedades atómicas para describir las interacciones proteína-proteína (PPI) entre dos pares de proteínas: no solo $Nav\alpha$ con subunidades $Nav\beta 1$ ($Nav\alpha/Nav\beta 1$) sino también $Nav\alpha$ con subunidades $Nav\beta 3$ ($Nav\alpha/Nav\beta 3$). El término PPI implica que ambos pares siempre se trataron en paralelo para las tres especies de mamíferos que proporciona una visión total y sistemática de los patrones quimiométricos. Si bien, las estructuras (3D) del primer par se han dilucidado experimentalmente (mediante microscopía crioelectrónica o cristalografía), no existe información estructural (3D) para el último par. Esto significa que, por un lado, estudiamos los complejos $Nav\alpha/Nav\beta 1$ existentes, mientras que por otro lado aplicamos directamente nuestros hallazgos para crear una interfaz hasta ahora desconocida (símbolo /) entre $Nav\alpha$ con $Nav\beta 3$ ($Nav\alpha/Nav\beta 3$). En adelante, tanto la interfaz observada como la postulada se denominarán IF, para abreviar. El ectodominio de dicha IF contiene las asas extracelulares de la subunidad $Nav\alpha$, que en nuestro estudio se denominarán ECL. Todos los segmentos ECL (cortos) en interacción con las subunidades

Nav β 1 o Nav β 3 se designarán como IF-ECL. En total, las ECL siempre pertenecen a una subunidad α , nunca a una subunidad β , porque el dominio extracelular de esta última carece de asas, solo hebras beta antiparalelas que se doblan en giros y horquillas para formar láminas beta ordenadas (dominio de inmunoglobulina, IgD). Finalmente, las IF-ECL abarcan segmentos de aminoácidos cortos, a veces solo unos pocos residuos individuales, que estudiamos a escala secuencial o incluso atómica.

3. JUSTIFICACIÓN

Se han identificado más de mil mutaciones puntuales en los Navs humanos, mientras que algunas de ellas se han asociado con trastornos neurológicos, cardiovasculares, musculares y psiquiátricos, como epilepsia, arritmia, parálisis muscular, síndrome de dolor y un amplio espectro en el trastorno del autismo. [Wolf y Mannion, 1999; Catterall, 2014; Claes *et al.*, 2009; Huang *et al.*, 2017; Rubinstein *et al.*, 2016]. Además, diversas toxinas de importancia clínica se han estudiado en la modificación del *gating* de los Navs. Como consecuencia, los Navs son blancos moleculares de una amplia variedad de toxinas naturales y fármacos terapéuticos clínicos [Gilchrist *et al.*, 2014; Catterall *et al.*, 2007; Bagal *et al.*, 2015; Ruiz y Krauz, 2015; Stevens *et al.*, 2011].

Investigaciones en electrofisiología y biología molecular, han avanzado en la exploración de un creciente número de enfermedades en los Navs llamadas “canalopatías”, en donde la utilización de técnicas como *patch clamp*, clonación molecular y sistemas bioinformáticos han identificado determinantes moleculares específicos, los cuales son puntos clave en la distorsión de la corriente de Na⁺ (Lehmann-Horn y Jurkat-Rott, 2003; Islas *et al.*, 2013; Scior *et al.*, 2015; Sánchez-Solano *et al.*, 2016).

Aunque las Nav α s son fundamentales para la conductancia de Na⁺, detección de voltaje y generación de potenciales de acción, Las Nav β s en interfaz con las Nav α s modulan la corriente de Na⁺ en diferentes estados del *gating* [Brackenbury e Isom, 2011].

Recientemente, han resuelto estructuras por Cryo-EM que resuelven la interfaz Nav α /Nav β [Shen *et al.*, 2017; Yan *et al.*, 2017; Pan *et al.*, 2018; Shen *et al.*, 2019; Jiang *et al.*, 2020].

Sin embargo, hasta la fecha se desconoce como las Nav β s ejercen la función mecánica y estructural a nivel atómico para realizar cambios en el *gating*.

En esta investigación se propone una solución por quimiometría al problema que subyace en la interfaz Nav α /Nav β en las diferentes isoformas de los Navs con tres especies de mamífero identificando patrones de interacción de residuos y una solución teórica de la función mecánica-estructural que ejercen Nav β 1 y Nav β 3 en las subunidades Nav α en la inactivación rápida del *gating*.

4. ANTECEDENTES

La extensa evidencia, indica que las subunidades Nav β modulan el gating de las Nav α , así como también alteran diversos efectos de fármacos al co-expresarse con las Nav α [cf. Brackenbury e Isom, 2011].

Pocos estudios o ninguno se han enfocado en el análisis quimiométrico del movimiento mecánico-estructural que las subunidades Nav β ejercen en la modulación de la corriente de Na $^+$. Sin embargo, estudios confirman que regiones específicas y residuos de las ECLs de las Nav α s en interfaz con las subunidades Nav β s son determinantes moleculares importantes involucrados en los cambios de la corriente de Na $^+$.

Diversos estudios indican que el dominio extracelular de IgD de las Nav β s interactúa con las Nav α en las ECLs.

Chen y Cannon, 1995, comprobaron que al eliminar 34 residuos en la región intracelular carboxi-terminal de Nav β 1, no son necesarios para la modulación del *gating* en Nav1.4 α

Se ha demostrado que anticuerpos específicos sitio-dirigidos a epítomos de las ECLs S5 y S6 de los DI y DIV bloquean la unión de la toxina α de escorpión [Tejedor y Catterall, 1988; Thomsen y Catterall, 1989].

Makita *et al.*, 1996, demostraron que Nav β 1 se une a las ECLs S5 y S6 de los DI y DIV de Nav1.4 α y están implicadas en la modulación del *gating* en la inactivación rápida por Nav β 1.

McCormick *et al.*, 1998 demostraron que determinantes moleculares específicos en Nav β 1 son necesarios para unión y modulación de Nav α ; el componente proteico del IgD, pero no el carbohidrato es necesario para la función de Nav β 1. Además, realizaron quimeras en donde demostraron que la eliminación del dominio intracelular de la subunidad Nav β 1 no es necesario para la modulación de la subunidad Nav1.2 α , sino la eliminación de segmentos del IgD extracelular.

Posteriormente, McCormick *et al.*, 1999 determinaron que el IgD de Nav β 1 es necesario y suficiente para modular la Nav α .

Se ha demostrado que el anestésico local lidocaína interrumpe la inactivación rápida en Nav α [Bennett *et al.*, 1995; Hille, 1977] e interactúa con los residuos Phe-1764 y Tyr-1771 en la ECL S6 DIV de Nav1.2 α [Bennett *et al.*, 1995; Ragsdale *et al.*, 1994]. Además, se ha propuesto que Nav β 1 y Nav β 3 se une a las ECLs S5 y S6 y dicha unión Nav α /Nav β 1 atenúa

el bloqueo por lidocaína pudiendo desestabilizar la unión de lidocaína a la ECL S6 DIV [Makita *et al.*, 1996; McCormick *et al.*, 1998; Qu *et al.*, 1999; Lenkowski *et al.*, 2003].

Los grupos de investigación del Dr. Eduardo Salinas-Stefanon del Laboratorio de Biofísica Cardíaca del Instituto de Fisiología y del Dr Thomas R. F. Scior del Laboratorio de Simulaciones Moleculares Computacionales de la Benemérita Universidad Autónoma de Puebla, han identificado determinantes moleculares en Nav β 1 que están implicados en la interfaz Nav α /Nav β 1 y en la modulación del gating de Nav1.4 α [Islas *et al.*, 2013; Scior *et al.*, 2015; Sánchez-Solano *et al.*, 2016].

Además, se ha resuelto experimentalmente por Cryo-EM la interfaz Nav α /Nav β 1 [Shen *et al.*, 2017; Yan *et al.*, 2017; Pan *et al.*, 2018; Shen *et al.*, 2019; Jiang *et al.*, 2020], pero no en Nav α /Nav β 3.

Namadurai *et al.*, 2015, mencionan que Nav β 1 y Nav β 3 se podrían unir a Nav α mediante el mismo mecanismo. Además, Ferrera *et al.* [2006], demostraron que Nav β 1 determina el entorno eléctrico del canal cambiando las cargas superficiales que afectan electrostáticamente la activación de los canales.

Zhu *et al.*, [2017], concluyeron de sus estudios *in vitro* que Nav β 1 y Nav β 3 aceleran la inactivación rápida al tener contacto con el S4 en Nav1.5 α , específicamente: Nav β 1 en S4 (DIII y DIV) y Nav β 3 en S4 DIII. Dichos estudios apoyan la base teórica en esta investigación, en la que un sensor de voltaje pudiera colisionar con una subunidad β desencadenando una serie de movimientos estructurales en la subunidad formadora del poro que aceleran la inactivación rápida.

Finalmente, los antecedentes mencionados brindan un fuerte soporte y evidencia de sitios de interacción para realizar estudios computacionales por modelado *in silico* en un enfoque quimiométrico de las interfaces Nav α /Nav β 1 y Nav α /Nav β 1.

5. OBJETIVOS

5.1. *General*

Identificar por métodos *in silico* la interacción funcional entre las subunidades Nav β (Nav β 1 y Nav β 3) y las isoformas de las subunidades Nav α en las tres especies de mamífero.

5.2. *Específicos*

- Realizar MSA de las isoformas de las Nav α y las subunidades Nav β de las tres especies de mamífero para identificar sitios PPI potenciales.
- Realizar modelos por homología de las isoformas de las Nav α y las subunidades Nav β de las tres especies de mamífero para identificar sitios PPI.
- Realizar un análisis quimiométrico estructural de la interfaz α/β para determinar patrones de interacción entre los modelos con las PPI identificadas.
- Realizar un análisis quimiométrico de interfaz entre las IF-ECLs de las isoformas de Nav α y las subunidades Nav β de las tres especies de mamífero para identificar similitudes estructurales entre isoformas y especies que elucidan patrones de interacción y obtener un panorama más amplio sobre las diferencias de interfaz entre isoformas y especies.

6. HIPÓTESIS

Los determinantes moleculares en la interfaz $Nav\alpha/Nav\beta$ determinan la modulación de la inactivación rápida de los canales de sodio dependientes de voltaje de mamífero.

7. MATERIAL Y MÉTODOS

7.1. Modelado por homología *in silico*

Se llevaron a cabo técnicas de modelado de homología para generar modelos de estructuras (3D) de las subunidades β hasta ahora desconocidas de ratones y ratas (mNav β 1, mNav β 3, rNav β 1 y rNav β 3) así como las isoformas de los siguientes complejos Nav α /Nav β : hNav1.1 α hNav1.3 α , hNav1.6 α , hNav1.8 α , hNav1.9 α , mNav1.1 α a mNav1.9 α , rNav1.1 α a rNav1.4 α , rNav1.6 α a rNav1.9 α . Se aplicaron dos programas, MODELLER 9.22 [Webb y Sali, 2014] y Chimera alpha V.1.14 [Pettersen *et al.*, 2004], utilizando las siguientes estructuras molde cryo-EM como (3D): hNav1.4 α /hNav β 1: cadenas A y B, PDB: 6AGF [Pan *et al.*, 2018], hNav1.2 α : cadena A, PDB 6J8E [Pan *et al.*, 2019], hNav1.7 α : cadena A, PDB 6J8G [Shen *et al.*, 2019], rNav1.5 α : PDB 6UZ0 [Jiang *et al.*, 2020]. Las siguientes estructuras determinadas por CRX también se tomaron como estructuras molde: hNav β 2, PDB 5FEB [Das *et al.*, 2016], hNav β 3: cadena A, PDB 4L1D [Namadurai *et al.*, 2014] y finalmente hNav β 4, PDB 4MZ2 [Gilchrist *et al.*, 2013].

7.2. Identificación de residuos que interactúan en la interfaz entre Nava y Nav β

El advenimiento de una estructura completa de canales de sodio con una unidad formadora de poro y proteínas auxiliares de un organismo superior (vertebrado), ha marcado el comienzo de una nueva área de análisis en la biología estructural para dar una idea de los fundamentos mecanicistas de la modulación de las subunidades Nav α .

La elucidación de la estructura fue un paso fundamental porque los canales Nav de vertebrados se componen de una proteína α monomérica (de cadena única) con cuatro dominios diferentes (I a IV), en contraste con los canales homotetraméricos conocidos hasta ahora de especies bacterianas sin proteínas auxiliares. Precisamente, nuestro trabajo quimioinformático explota esta primera interfaz observada experimentalmente del complejo de isoformas Nav1.4 α de una especie de vertebrados: la anguila (eeNav α /Nav β , PDB: 5XSY [Yan *et al.*, 2017]). Esta estructura molde ayudó a analizar los aminoácidos que interactúan en la interfaz PPI en el ECR (eeNav α /Nav β) aplicando herramientas de software para detectar contactos en la interfaz y las bases de datos de rotámeros Dunbrack y Dynameomics [Shapovalov y Dunbrack 2011; Scouras y Daggett, 2011]. El MSA siempre se realizó utilizando Clustal Omega 1.2.4 basado en la web con su configuración predeterminada [Clustal Omega, 2008; [Clustal Omega < Multiple Sequence Alignment < EMBL-EBI](#)]. Para

eliminar los problemas de numeración de residuos a la vista de longitudes de secuencia variables, cada residuo identificado que esta "interactuando" (es decir, formando el PPI de eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017]) se denominó como un pequeño segmento de siete aminoácidos adyacentes. En su posición central, el residuo que interactúa está flanqueado por tres aminoácidos a cada lado. El patrón esquemático es "yyyXyyy", donde "y" simboliza cualquier aminoácido, mientras que "X" es el aminoácido que interactúa.

7.3. Modelado in silico de estructuras molde para los modelos de PPI

Para generar las interfaces se utilizaron las siguientes estructuras molde: la estructura cryo-EM de hNav β 1, PDB 6AGF [Pan *et al.*, 2018], la cadena A de la estructura cristalina de hNav β 3, PDB 4L1D [Namadurai *et al.*, 2014]. Además, se generaron los siguientes modelos por homología para las subunidades mNav β 1, mNav β 3, rNav β 1 y rNav β 3, así como de los complejos de las isoformas computadas hNav1.1 α , hNav1.3 α , hNav1.5 α , hNav1.6 α , hNav1.8 α , hNav1.9 α , mNav1.1 α a mNav1.9 α , rNav1.1 α a rNav1.4 α , rNav1.6 α a rNav1.9 α . También se utilizaron las estructuras cryo-EM de la isoforma hNav1.4 α con cadenas A y B, PDB 6AGF [Pan *et al.*, 2018]), hNav1.2 α con la cadena A, PDB 6J8E [Pan *et al.*, 2019]), hNav1.7 α con la cadena A, PDB 6J8G [Shen *et al.*, 2019]), rNav1.5 α , PDB 6UZ0 [Jiang *et al.*, 2020]). Cabe aclarar que no se realizaron combinaciones quiméricas entre especies.

7.4. Determinación de las regiones extracelulares de las Navas

Se determinaron las estructuras y secuencias de las ECL de las isoformas hNav1.1 α a hNav1.9 α , mNav1.1 α a mNav1.9 α y rNav1.1 α a rNav1.9 α , en base a las estructuras molde disponibles: eeNav1.4 α , hNav1.2 α , hNav1.4 α , rNav1.5 α y hNav1.7 α , que se habían descargado de la base de datos Orientations of Proteins in Membranes (OPM) [Lomize *et al.*, 2006]. OPM proporciona información espacial sobre el empaquetamiento de la bicapa lipídica de la parte helicoidal transmembrana para nuestros modelos de canal es. OPM ayudó a definir el ECR, es decir, en qué posición sobresale el bucle y vuelve a entrar en TMH.

7.5. Cálculo de las propiedades de las ECLs

Se calcularon los 432 segmentos de aminoácidos, que definen todas las ECL bajo nuestro escrutinio, es decir, D1 a DIV con 4 asas del ectodominio, y cada uno de ellos por nueve isoformas para tres especies produce un total de $16 \times 9 \times 3 = 432$ modelos topológicos.

Se "extrajeran" de las secuencias primarias de hNav1.1 α a hNav1.9 α , mNav1.1 α a mNav1.9 α y rNav1.1 α a rNav1.9 α . Las longitudes de ECL y las propiedades informativas de los aminoácidos que interactúan, las cuales se calcularon y documentaron sistemáticamente para los análisis de PPI posteriores. La gran cantidad de datos hizo que la creación de secuencias de comandos fuera un activo muy valioso en Chimera (consulte las secuencias de comandos al final de SM). Las propiedades quimiométricas (descriptores o parámetros) incluyen áreas polares, no polares, cisteínas y aminoácidos aromáticos. Cabe destacar que se observó y documentó el enlace de hidrógeno, pero se omitió la presentación de datos, ya que la construcción de la red de hidrógeno intra o intermolecular es una opción estándar. Los enlaces H en las interfaces se muestran fácilmente, es decir, redes de hidrógeno intermoleculares (en la zona de contacto, que se define por un radio de selección de átomos) entre dos proteínas (c. f. modelo 3D de hNav1.4 α /Nav β 3 y la entrada de PDB 6AGF [Pan *et al.*, 2018] con hNav1.4 α /Nav β 1 en formato PDB en SM).

7.6. Cálculo de las propiedades químicas en la superficie de las IF-ECL

Para todos los modelos 3D, las energías potenciales de las estructuras se minimizaron y las cargas totales y parciales se agregaron con un radio de sonda de agua de 1,4 Å y una densidad de vértice de 2,0 [Pedretti *et al.*, 2004]. El volumen molecular, el SAA total y el SAA polar y no polar de los átomos cargados o no cargados se estimaron en Å² para todas las ECLs [Pettersen *et al.*, 2004]. Habitualmente, todos los datos se calcularon para las nueve isoformas de las tres especies.

7.7. Interacciones electrostáticas de las superficies IF-ECL (MEPS)

Las estructuras de los archivos de entrada (modelos 3D) se prepararon con el complemento PDB2PQR en Chimera, de acuerdo al protocolo [Dolinsky *et al.*, 2007]. El cuál, asigna radios y cargas específicas en el soporte estructural para los cálculos electrostáticos por Poisson-Boltzmann (APBS), que constituye un nivel teórico más alto que las fuerzas electrostáticas calculadas en base al solucionador simple de ecuaciones de Coulomb [Baker *et al.*, 2001]. La salida numérica se convirtió para la visualización gráfica de MEPS. Para ello, se excluyeron todos aquellos puntos de datos por encima y por debajo de un determinado umbral (± 30) (protocolo empírico). Los valores de fuerza electrostática en el rango de +30 a -30 se consideraron para el escalamiento lineal del código de color entre

+1 y -1 (normalización de datos de la unidad uno). La carga correspondiente en cada vértice de la superficie se asignó utilizando la interfaz UCSF Chimera alpha v. 1.14 [Pettersen *et al.*, 2004] en selecciones moleculares para cada ECL.

7.8. Cálculo de propiedades de interfaz de Nava/Nav β 1 y Nava/Nav β 3

En la interfaz, el área enterrada, el SAA total y las áreas polares y no polares se calcularon en Å² aplicando scripts en Chimera alpha v. 1.14 [Pettersen *et al.*, 2004]. Para una comparación directa, algunos valores se expresaron como porcentajes para reflejar la porción relativa (%) de la longitud total del asa (100%) para tener en cuenta la enorme variación en la longitud. Además, se tomaron como base residuos con cargas positivas o negativas para comparar el MEPS en la interfaz.

7.9. Validación del modelo externo de los modelos PPI

En una visión más general, el advenimiento del conocimiento estructural sobre el proteoma celular ha marcado el comienzo de una nueva área de estudios de PPI que identifican “hot spots” de interacción entre proteínas adyacentes [Sarvagalla *et al.*, 2016]. Una vez finalizado nuestro estudio, se llevó a cabo una generación de interfaz totalmente automatizada. La herramienta basada en la web identificó los mismos residuos que interactúan (ver la sección final de los Materiales Suplementarios [Cukuroglu *et al.*, 2011]).

8. RESULTADOS

8.1. *Determinación de las PPI en las isoformas de los Navs*

Los patrones de propiedades de los residuos observados y calculados que interactúan en la interfaz se describieron para ocho números Id de PPI (a continuación, llamados PPI-Id) por nueve isoformas por especies para dos pares ($\text{Nav}\alpha/\text{Nav}\beta 1$ y $\text{Nav}\alpha/\text{Nav}\beta 3$), los cuales produce 832 PPI en 27 modelos 3D ($8 \times 9 \times 3 \times 2$) basados en estructuras 3D conocidas y modeladas *in silico*. Cuando se tiene en cuenta los cuatro dominios de cada subunidad α , la cadena de poros y el hecho de que cada uno de los cuatro dominios (D1 a DIV) expone cuatro ECLs, entonces la gran cantidad de 432 conjuntos de cálculos ($16 \text{ ECLs} \times 9 \text{ isoformas} \times 3 \text{ especies}$) fueron elaborados, realizados, recolectados, documentados e interpretados. La Tabla 1 proporciona una sinopsis de los resultados obtenidos que describen la interacción entre $\text{Nav}\alpha/\text{Nav}\beta 1$ o $\text{Nav}\alpha/\text{Nav}\beta 3$. Es de destacar que se identificaron ocho patrones de interacción polar computados diferentes en los cuatro dominios (DI a DIV). Los cuales extienden bastante la literatura existente por su análisis sistemático y completo.

Tabla 3. Sinopsis de los patrones de PPI estudiados.

Los sitios de interacción están etiquetados como PPI-Id con números arábigos del "1" al "8" para su identificación. Los patrones resultantes están etiquetados con números romanos de "I" a "IX". Ciertos PPI entre $Na_v\alpha$ / $Na_v\beta 1$ y $Na_v\alpha$ / $Na_v\beta 3$ se han observado experimentalmente mediante la elucidación de las estructuras. Sus respectivas entradas PDB y valores PPI-Id están marcados en negrita. Se utilizaron como moldes para el recordatorio. Los valores Y/N en las celdas de la tabla simbolizan SÍ/NO en referencia a la presencia/ausencia de contribuciones a PPI.

Isoforma	IF-ECLs ⁶	S5 DI		S1-S2 DIII		S5 DIV	S6 DIV			PPI Patrón
	PPI-Id ³	1	2	3	4	5	6	7	8	
$hNa_v1.1\alpha^5$	P35498 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$mNa_v1.1\alpha^5$	A2APX8 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$rNa_v1.1\alpha^5$	P04774 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$hNa_v1.2\alpha^4$	6J8E ¹	Y	Y	Y	Y	Y	Y	Y	Y	II
$mNa_v1.2\alpha^5$	B1AWN6 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$rNa_v1.2\alpha^5$	P04775 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$hNa_v1.3\alpha^5$	Q9NY46 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$mNa_v1.3\alpha^5$	A2ASI5 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$rNa_v1.3\alpha^5$	P08104 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$eeNa_v1.4\alpha^4$	5XSY ¹	N	Y	Y	Y	Y	Y	Y	Y	IX
$hNa_v1.4\alpha^4$	6AGF ¹	Y	Y	Y	Y	Y	Y	Y	Y	II
$mNa_v1.4\alpha^5$	Q9ER60 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$rNa_v1.4\alpha^5$	P15390 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$hNa_v1.5\alpha^5$	Q14524 ²	Y	Y	N	Y	Y	N	Y	Y	III
$mNa_v1.5\alpha^5$	Q9JIV9 ²	Y	Y	N	Y	Y	N	Y	Y	III
$rNa_v1.5\alpha^4$	6U70 ¹	Y	Y	N	Y	Y	N	Y	Y	III
$hNa_v1.6\alpha^5$	Q9UQD0 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$mNa_v1.6\alpha^5$	Q9WTU3 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$rNa_v1.6\alpha^5$	O88420 ²	Y	Y	Y	Y	Y	Y	Y	Y	II
$hNa_v1.7\alpha^4$	6J8G ¹	Y	Y	Y	Y	Y	N	Y	Y	I
$mNa_v1.7\alpha^5$	Q62205 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$rNa_v1.7\alpha^5$	O08562 ²	Y	Y	Y	Y	Y	N	Y	Y	I
$hNa_v1.8\alpha^5$	Q9Y5Y9 ²	Y	Y	Y	N	Y	N	Y	Y	IV
$mNa_v1.8\alpha^5$	Q6QIY3 ²	Y	Y	N	N	Y	N	Y	Y	V
$rNa_v1.8\alpha^5$	Q62968 ²	Y	N	N	N	Y	N	Y	Y	VI
$hNa_v1.9\alpha^5$	Q9UI33 ²	Y	N	Y	N	Y	N	N	N	VII
$mNa_v1.9\alpha^5$	Q9R053 ²	Y	Y	N	N	Y	N	N	N	VIII
$rNa_v1.9\alpha^5$	O88457 ²	Y	Y	N	N	Y	N	N	N	VIII

¹PDB (<http://www.rcsb.org/>); ²UniProt (<https://www.uniprot.org/>); ³PPI-Id para las interacciones computadas polares de las subunidades $hNa_v\alpha$ y $hNa_v\beta$; ⁴estructuras determinadas por modelado por homología (3D templates); ⁵models; ⁶ECLs que forman interfaces $Na_v\beta 1$ o $Na_v\beta 3$ son llamadas IF-ECLs; Y: interacción; N: No interacción; $eeNa_v1.4\alpha$: *Electrophorus electricus* $Na_v1.4\alpha$; I to XI: Grupos de patrones PPI.

8.1.1. Análisis PPI del complejo estructural eeNav1.4 α /eeNav β 1

La Figura 5 presenta las PPI en eeNav1.4 α /eeNav β 1 (PDB: 5XSY [Yan *et al.*, 2017]). En donde ya se habían publicado seis interacciones en tres ECL antes de este estudio: S1-S2 DIII, S5 DIV y S6 DIV. Sin embargo, el análisis reveló un sitio de interacción hasta ahora inédito en ECL S5 DI (6 + 1 = 7 PPI-Ids).

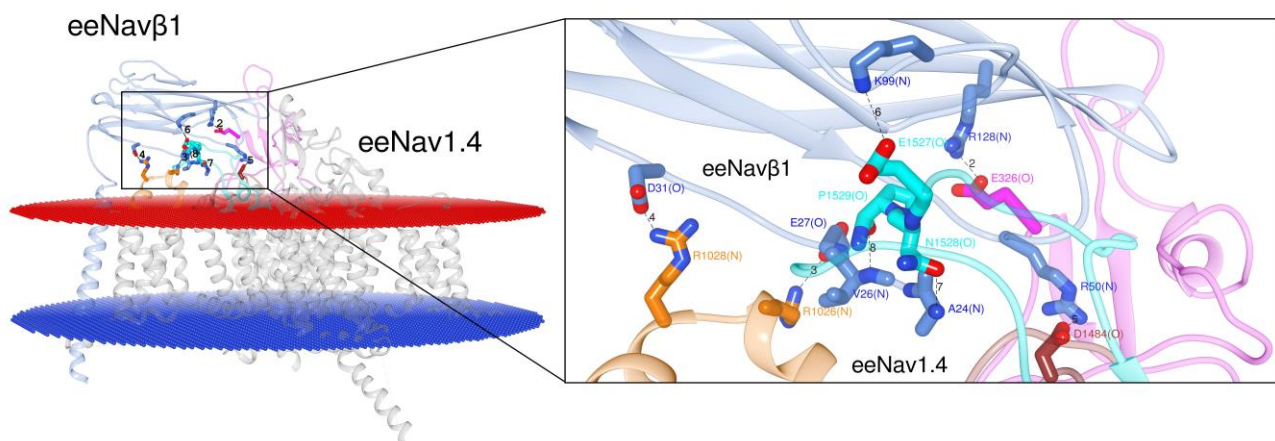


Figura 5. Visualización del modelo 3D con vista desde el ECL S6 DIV de eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017].

El cuadro muestra detalles sobre los residuos que interactúan en la interfaz. Etiquetas 2 a 8: números de identificación PPI (PPI-Ids) de interacciones polares calculadas (consulte Tabla S1, Materiales suplementarios); los aminoácidos están marcados por códigos de una letra con sus números de residuo de secuencia primaria y átomos que interactúan, e. g. A24(N). Colores: límites de la membrana extracelular (rojo oscuro), límites de la membrana intracelular (azul marino), regiones proteicas transmembrana e intracelular de Nav α que no participan en PPI (gris), subunidad Nav β 1 (azul cian); S5 DI (magenta), S1-S2 DIII (naranja), S5 DIV (marrón), S6 DIV (cian); interacciones polares calculadas: líneas punteadas negras. Visualización lograda por Chimera Alpha 1.14 [Pettersen *et al.*, 2008].

8.1.2. Análisis PPI de los modelos hNav1.4 α /hNav β 1 y hNav1.4 α /hNav β 3

Los complejos molde 3D eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017] y hNav1.4 α /hNav β 1 [Pan *et al.*, 2018] poseen una alta identidad de secuencia (Nav α \approx 65% y Nav β \approx 46%, respectivamente) además de un alto grado de residuos conservados por homología. Una medida de la deformación geométrica es la denominada desviación cuadrática media (RMSD). Las entradas de PDB 5XSY [Yan *et al.*, 2017] versus 6AGF [Pan *et al.*, 2018] se compararon en términos de RMSD para ambas subunidades: Nav α \approx 0,942, Nav β \approx 0,955. Una primera inspección visual de ambas estructuras molde 3D también reveló qué tan bien conservadas están todas las IF-ECL con Nav β 1 entre ambas especies.

Después de la detección del séptimo sitio PPI (véase 8.1.1.), Inspeccionamos la estructura molde 3D hNav1.4 α /hNav β 1 [Pan *et al.*, 2018] y el modelo por homología hNav1.4 α /hNav β 3 [Pan *et al.*, 2018; Namadurai *et al.*, 2014] (Figura 3). En esta etapa detectamos otro sitio PPI en el ECL S5 DI. A continuación, etiquetamos los seis sitios PPI

publicados y los dos detectados de la siguiente manera: PPI-ID 3 a 8 y PPI-ID 1 y 2, respectivamente. Es de destacar que los PPI-Ids 3 a 8 se observaron en la estructura molde 3D eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017]. Sin embargo, la rotación de la cadena lateral por la biblioteca de rotámeros incorporada de Chimera [Pettersen *et al.*, 2008, Scouras *et al.*, 2011, Shapovalov *et al.*, 2011] tuvo que aplicarse en la estructura molde 3D hNav1.4 α /hNav β 1 [Pan *et al.*, 2018]. Ambos PPI-ID detectados (1 y 2) se encuentran en la ECL S5 DI.

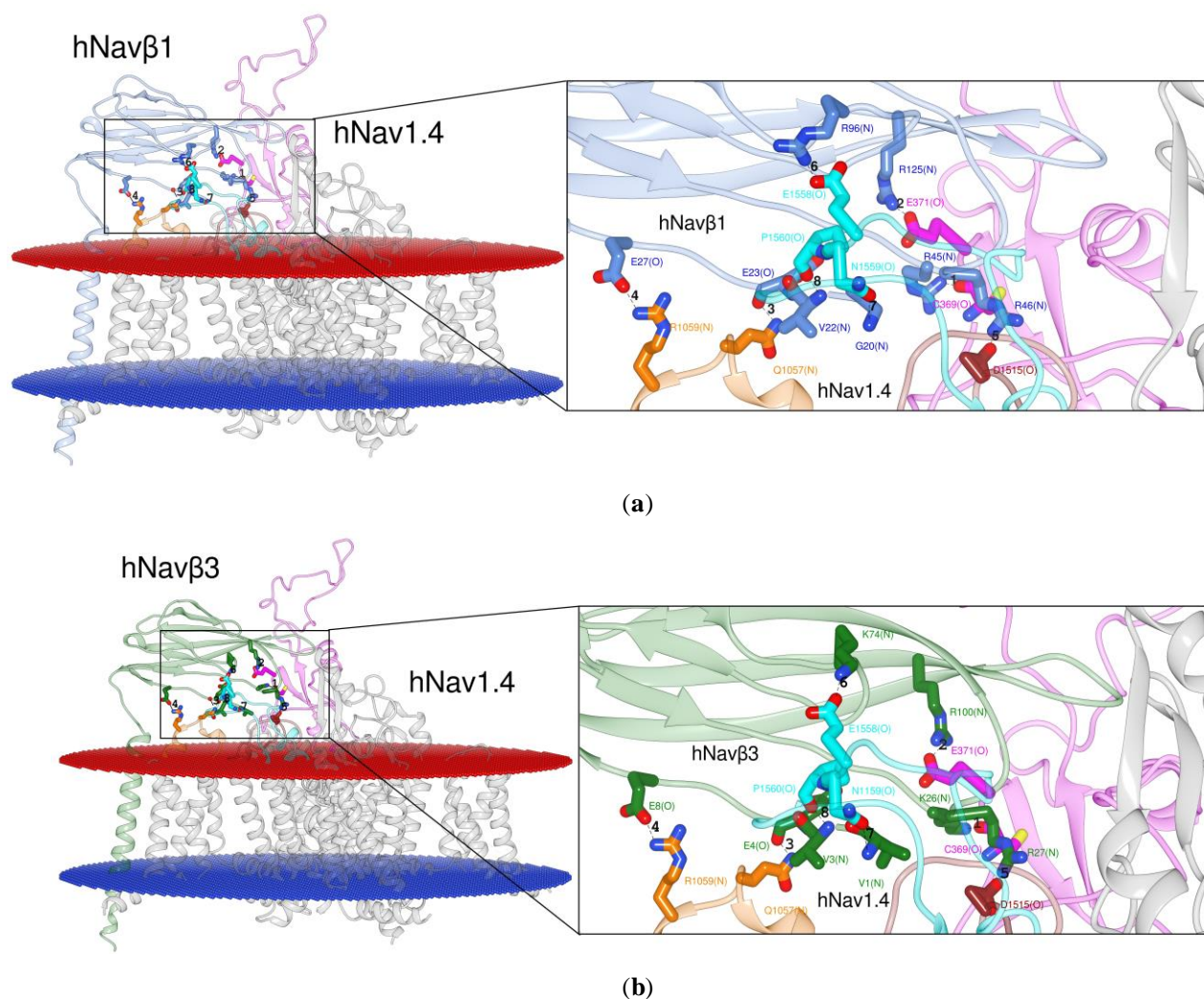


Figura 6. Visualización de los modelos 3D con vista desde el ECL S6 DIV de hNav1.4 α /hNav β 1 [Pan *et al.*, 2018].

Modelos por homología 3D de (a) hNav1.4 α /hNav β 1, basado en [Pan *et al.*, 2018] y (b) hNav1.4 α /hNav β 3, basado en [Pan *et al.*, 2018; Nama dura i *et al.*, 2014]. Se presentan detalles atómicos en la interfaz. Etiquetas 1 a 8: PPI-Ids de interacciones polares calculadas (Tabla S1, Materiales suplementarios); los aminoácidos están etiquetados por códigos de una letra con sus números de residuo de secuencia primaria y átomos que interactúan entre paréntesis, e. g. en la parte inferior D1515 (O). Colores: límites de la membrana extracelular (rojo oscuro), límites de la membrana intracelular (azul marino), regiones proteicas transmembrana e intracelular de Nava que no participan en PPI (gris), subunidad Nav β 1 (azul cian), subunidad Nav β 3 (verde bosque), S5 DI (magenta), S1-S2 DIII (naranja), S5 DIV (marrón), S6 DIV (cian); interacciones polares calculadas: líneas punteadas negras. Visualización hecha con Chimera Alpha 1.14 [Pettersen *et al.*, 2008].

8.1.3. *Identificación de los residuos en las Navas*

El alineamiento de secuencias múltiples (MSA) fue realizado por el servidor Clustal Omega basado en web [Clustal Omega [Clustal Omega < Multiple Sequence Alignment < EMBL-EBI](#)] con las secuencias primarias de eeNav1.4 α , hNav1.1 α a hNav1.9 α , mNav1.1 α a mNav1.9 α , rNav1.1 α a rNav1.9 α y todas las subunidades β (eeNav β 1, hNav β 1 a hNav β 4, mNav β 1 a mNav β 4 y rNav β 1 a rNav β 4). Identificamos los aminoácidos conservados de acuerdo a los residuos identificados por homología (cf. 8.1.1., 8.2.2), ya que constituyen componentes fundamentales estructurales y funcionalmente para el canal (Tabla 4). Como activo más valioso, los residuos interactivos conocidos hasta ahora de las estructuras molde 3D sirvieron como referencia para identificar los residuos en los demás modelos (Tabla 1 y Tabla S1). Las estructuras molde 3D fueron las siguientes: eeNav1.4 α /eeNav β 1, hNav1.4 α /hNav β 1 y hNav1.4 α /hNav β 3 (Figuras 5 y 6).

En Nav α , se detectaron dos residuos conservados en las posiciones de secuencia 1 y 5 en las ECL S5 DI y S5 DIV, respectivamente. En el apartado de la Tabla 4 de la subunidad β , las subunidades Nav β 1 y Nav β 3 contienen residuos parcialmente conservados en las posiciones de los PPI-Ids 2b, 3b y 5b, pero en el PPI-Id 8b la valina (V) permanece sin cambios. Curiosamente, ambas β también presentan dos residuos homólogos al mantener sus respectivas cargas en las posiciones 4 (negativa) y 6 (positiva) en todas las isoformas y especies. Por el contrario, la Tabla 4 también revela que los residuos equivalentes en Nav β 2 o Nav β 4 no se conservan cuando se comparan con ubicaciones en las tres secuencias de referencia de los moldes de eeNav β 1, hNav β 1 o hNav β 3.

A continuación, se describen dos casos de PPI-Id para ilustrar cómo se combinan las Tablas 3, 4 y S1 con las Figuras 2, 3 y S1 a S9. Tome la esquina superior izquierda de la Tabla 1. El valor en la celda para el PPI-Id 1 de Nav1.1 α es "Y", es decir, sí, hay un PPI. Esto corresponde a la cisteína (C) en la Tabla S1 (entrada de la primera fila de datos cruzada por la tercera columna: "agqCpeg (O)". Esto corresponde a la cadena "agqCpEgym" que fue generada por el MSA en la Tabla 4. La Tabla S1 informa que cisteína (C) está en contacto con arginina (R) de Nav β 1 humana (3^{ro} columna bajo el PPI-Id "1b" el valor "R" en la cadena "sckRrse (N)"). En este caso, la configuración espacial se representa en detalles atómicos (Figura S1). En este caso, el átomo de oxígeno de la amida (>C = O) de la cisteína forma un enlace de hidrógeno con un nitrógeno de la arginina en β .

Tabla 4. Alineamiento de secuencia múltiple de los PPI-Id.

El alineamiento de secuencia múltiple de los PPI-Id para las nueve subunidades α y las cuatro subunidades β de tres organismos mamíferos incluyendo la estructura molde 3D de anguila. El MSA identificó ocho residuos conservados u homólogos en Nava y Nav β . Asteriscos (*) en la primera columna: estructuras molde 3D de PDB. Letras mayúsculas en negrita: los ocho residuos alineados. Letras minúsculas: aminoácidos vecinos de los ocho residuos alineados para una identificación sin considerar números de secuencia. Colores: residuos cargados positivos y negativos (en azul y rojo, respectivamente), residuos polares y no polares (en cian y naranja, respectivamente).

Nava	UC ¹	ECLs			
		S5 DI 1, 2	S1-S2 DIII 3, 4	S5 DIV 5	S6 DIV 6,7,8
hNav1.1 α	P35498	agqCpEgym	yidQrKtik	gidDmfn	pnkVNPgss
mNav1.1 α	A2APX8	agqCpEgym	yidQrKtik	gidDmfn	pnkVNPgss
rNav1.1 α	P04774	agqCpEgym	yidQrKtik	gidDmfn	pnkVNPgss
hNav1.2 α	Q99250	agqCpEgyi	yieQrKtik	gidDmfn	pdkDHPgss
mNav1.2 α	B1AWN6	agqCpEgyi	yieQrKtikd	gidDmfn	pekDHPgss
rNav1.2 α	P04775	agqCpEgyi	yieQrKtik	gidDmfn	pekDHPgss
hNav1.3 α	Q9NY46	agqCpEgyi	yieQrKtik	gidDmfn	ptdIHPgss
mNav1.3 α	A2ASI5	agqCpEgyi	yieQrKtik	gidDmfn	pdaIHPgss
rNav1.3 α	P08104	agqCpEgyi	yieQrKtik	gidDmfn	pdaIHPgss
*ceNav1.4 α	P02719	agkCpEgyt	yiwRrRvik	gvdDifn	pdvENPgtg
*hNav1.4 α	P35499	aghCpEgye	yieQrRvir	gidDmfn	pnLENPgtg
mNav1.4 α	Q9ER60	aghCpEgye	yieQrRvir	gidDmfn	ptLENPgtg
rNav1.4 α	P15390	aghCpEgye	yieQrRvir	gidDmfn	ptLENPgtg
hNav1.5 α	Q14524	agtCpEgyr	yleErKtik	gidDmfn	ptlPNSngs
mNav1.5 α	Q9JJV9	agtCpEgyr	yleErKtik	gidDmfn	pnlPNSngs
rNav1.5 α	P15389	agtCpEgyr	yleErKtik	gidDmfn	pnlPNSngs
hNav1.6 α	Q9UQD0	agqCpEgyq	yieQrKtir	gidDmfn	ldkEHPgsg
mNav1.6 α	Q9WTU3	agqCpEgfq	yieQrKtir	gidDmfn	ldkEHPgsg
rNav1.6 α	O88420	agqCpEgfq	yieQrKtir	gidDmfn	ldkEHPgsg
hNav1.7 α	Q15858	sgqCpEgyt	yieRkKtik	ginDmfn	pkkVHPgss
mNav1.7 α	Q62205	sgqCpEgye	yieKkKtik	ginDmfn	pkkVHPgss
rNav1.7 α	O08562	sgqCpEgyi	yieKkKtik	ginDmfn	pkkVHPgss
hNav1.8 α	Q9Y5Y9	sgHcPdyi	yldQkPtvk	gidDmfn	pnlPNSngt
mNav1.8 α	Q6QIY3	aghCpNdyv	yleEkPrvk	gidDmfn	pnrPNSngt
rNav1.8 α	Q62968	aghCpGgyv	yleEkPrvk	gidDmfn	pnlPNSngs
hNav1.9 α	Q9UI33	nsaCslqye	hleNqPkiq	gidDifn	rskESCnss
mNav1.9 α	Q9R053	rsCpDgst	nlpSrPqve	gidDifn	eskASCnss
rNav1.9 α	O88457	srpCpNgst	nlpSrPqve	gidDifn	eakEHCnss
Nav β	UC ¹	7, 8, 3, 4	1,5	6	2
*ceNav β 1	A0A1L3MZ94	sngAcVEvdsDtea	sckMRgev	mgsKntf	yfdRtlit
*hNav β 1	Q07699	acgGcVEvdsEtea	sckRRset	ngsRgtk	hvyRllf
mNav β 1	P97952	awgGcVEvdsDtea	sckRRset	ngsRgtk	hvyRllf
rNav β 1	Q00954	awgGcVEvdsEtea	sckRRset	ngsRgtk	hvyRllf
*hNav β 3	Q9NY72	cfpVcVEvpsEtea	scmKReev	ngsKdlq	nvsRefe
mNav β 3	Q8BHK2	cfpVcVEvpsEtea	scmKReev	ngsKdlq	nvsRefe
rNav β 3	Q9JK00	cfpVcVEvpsEtea	scmKReev	ngsKdlq	nvsRefe
hNav β 2	O60939	grsMeVTlvpTlnv	fnsCYtvn	sgnPsky	yimNppd
mNav β 2	Q56A07	grsMeVTlvpTlsv	fnsCYtvn	sgnPsky	yitNppd
rNav β 2	P54900	grsMeVTlvpTlsv	fnsCYtvn	sgnPsky	yitNppd
hNav β 4	Q81WT1	sleVsVGkatDiya	fssCFgfe	vgsTkek	hvkNpke
mNav β 4	Q7M729	sleVsVGkatTiya	fssCYgfe	egsTkek	fvrNpke
rNav β 4	Q7M730	sleVsVGkatTiya	fssCYgfe	egsTkek	fvrNpke

¹: UC: Código UniProt: <https://www.uniprot.org>

La Tabla S1 informa acerca de esta instancia de PPI de una manera no gráfica. La IF tiene dos lados con PPI-Id 1 y PPI-Id 1b. En la subunidad formadora del poro de humano de la isoforma 1.1 (hNav1.1 α), el valor de cadena "agqCpeg (O)" informa que el residuo que interactúa es cisteína (C). Interactúa a través de su estructura de oxígeno átomo (O). En el

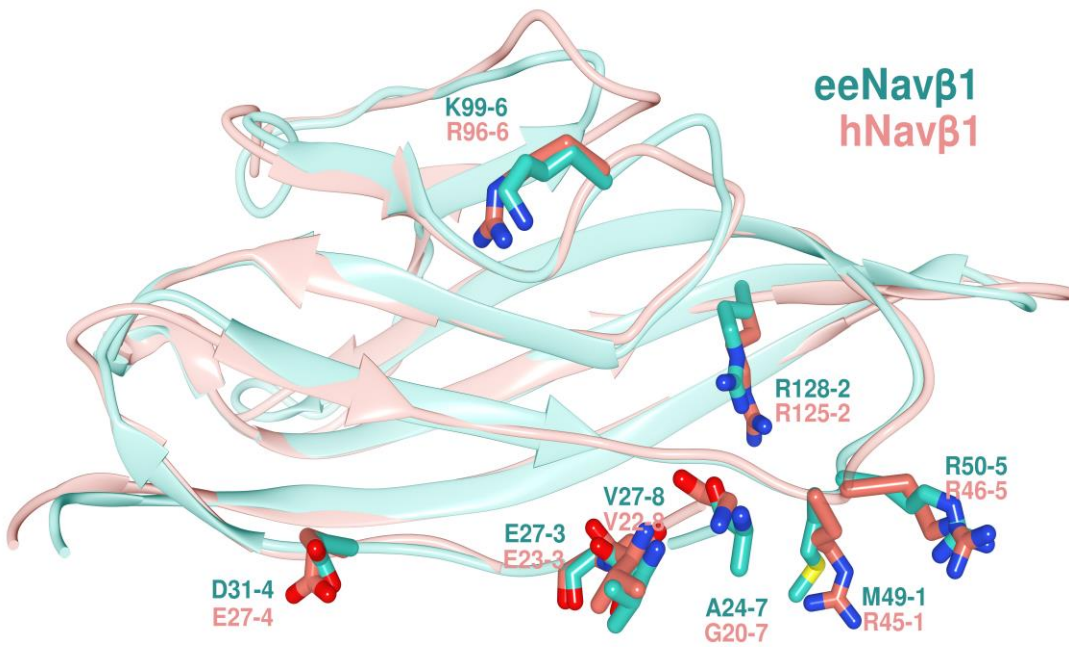
otro lado, la Tabla S1 contiene el valor de cadena "sckRrse (N)" en la posición PPI-Id "1b" para Nav β 1. Esto significa que la arginina es la contraparte. En una escala atómica, el átomo de nitrógeno del grupo guanidinio monocatiónico de la cadena lateral puede establecer el enlace de hidrógeno con una fuerte atracción polar.

En la Tabla 3, en la esquina inferior derecha se informa que no existe (valor "N") para PPI-Id 8 en el caso de rNav1.9 α . Nuevamente, la Tabla S1 brinda información no solo a nivel molecular sino también a escala atómica. La subunidad α de la isoforma 1.9 tiene una contribución potencial de una cisteína en la posición de secuencia PPI-Id 8 ("kehCnss"), pero las subunidades (Nav β) presentan una valina (V) que no interactúa. Este residuo alifático se conserva en las cuatro subunidades β . Como resultado directo, la Tabla 3 resume esta interacción negativa con un calificador "N" en su celda de tabla correspondiente junto con todas las demás contribuciones de "Sí" o "No" para las combinaciones sistemáticas de tres especies de mamíferos, nueve isoformas y ocho sitios potenciales de PPI (etiquetados PPI-ID 1 a 8). Las dos ilustraciones detalladas subrayan la riqueza informativa de los estudios quimiométricos para complementar los datos experimentales limitados. Sin embargo, también plantean los desafíos profesionales del modelado molecular en relación a la gestión y presentación de datos en grandes volúmenes.

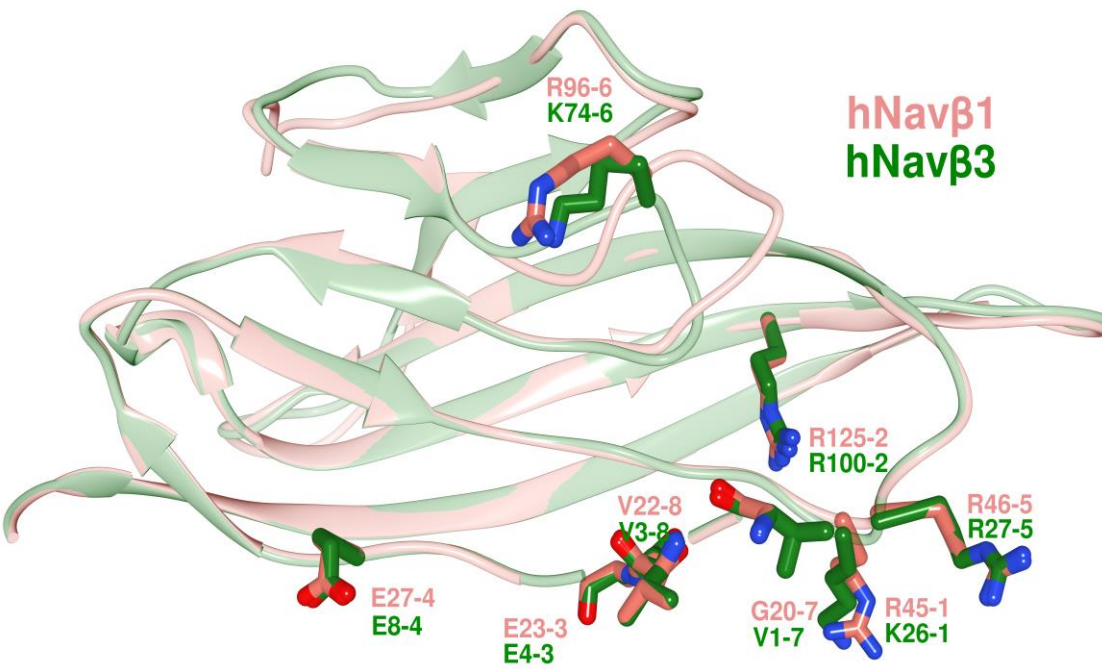
8.1.4. Alineación estructural de las Nav β s

La Figura 7 muestra los resultados del alineamiento estructural 3D (SA), es decir, las superposiciones del ectodominio (IgD) en el espacio para las estructuras molde 3D de las estructuras cryo-EM de eeNav β 1, hNav β 1. Además, se muestra el SA de las estructuras determinadas por CRX de hNav β 2 a hNav β 4 (PDB: 5XSY [Yan *et al.*, 2017], 6AGF [Pan *et al.*, 2018], 5FEB [Das *et al.*, 2016], 4L1D [Namadurai *et al.*, 2014] y 4MZ2 [Gilchrist *et al.*, 2013]).

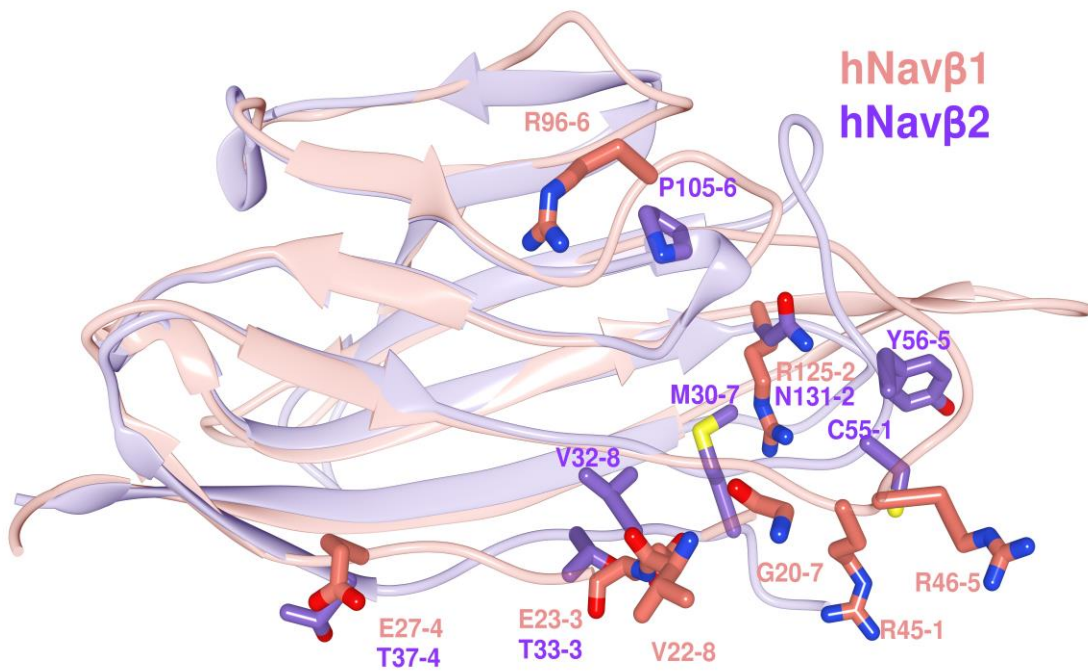
El SA de las subunidades eeNav β 1 y hNav β 1 proporciona un medio visual para identificar los residuos que interactúan para eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017] (Figura 2). Como puede observarse, las características esenciales permanecen en estrecha proximidad espacial mientras mantienen sus propiedades, excepto por uno de los ocho sitios de interacción: PPI-Id 1 (Figura 7a).



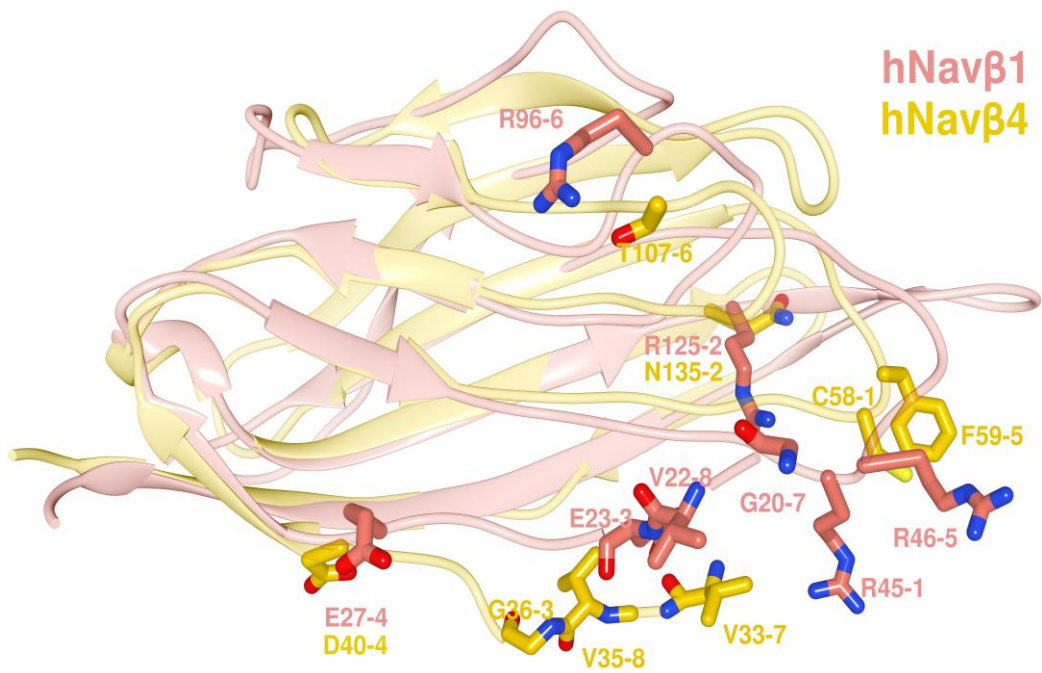
(a)



(b)



(c)

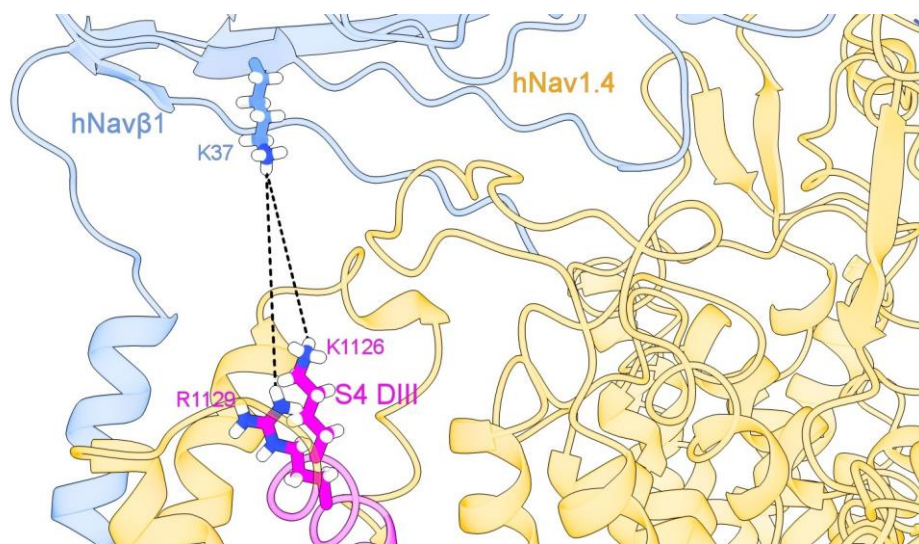


(d)

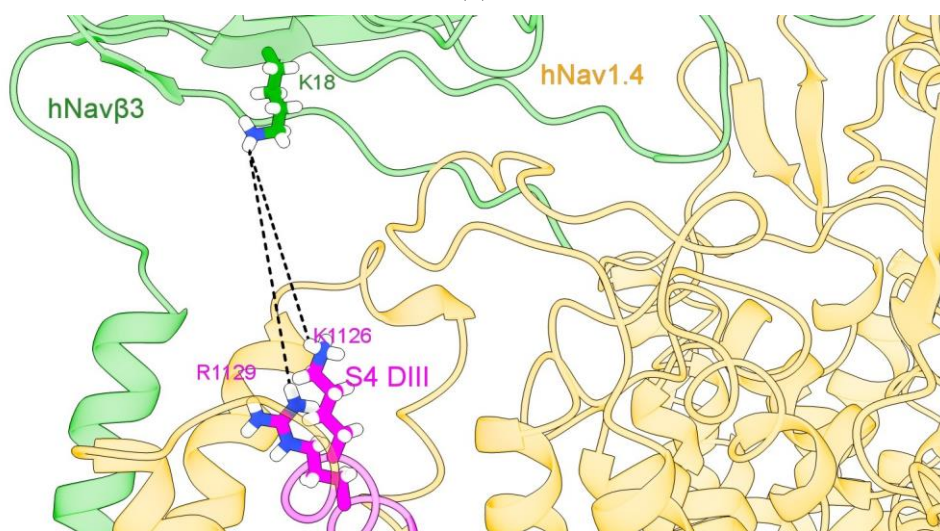
Figura 7. Alineamiento estructural de los ectodominios (IgD) de las estructuras molde de Navβs. (a) eeNavβ1 con hNavβ1; (b) hNavβ1 con hNavβ3; (c) hNavβ1 con hNavβ2; (d) hNavβ1 con hNavβ4; eeNavβ1: verde mar; hNavβ1: salmón; hNavβ3: verde bosque; hNavβ2: púrpura; hNavβ4: dorado; etiquetas 1 a 8: posiciones de los residuos de MSA de acuerdo a la Tabla 4.

8.2. Modulación de la aceleración de la inactivación rápida del gating del poro por Nav β

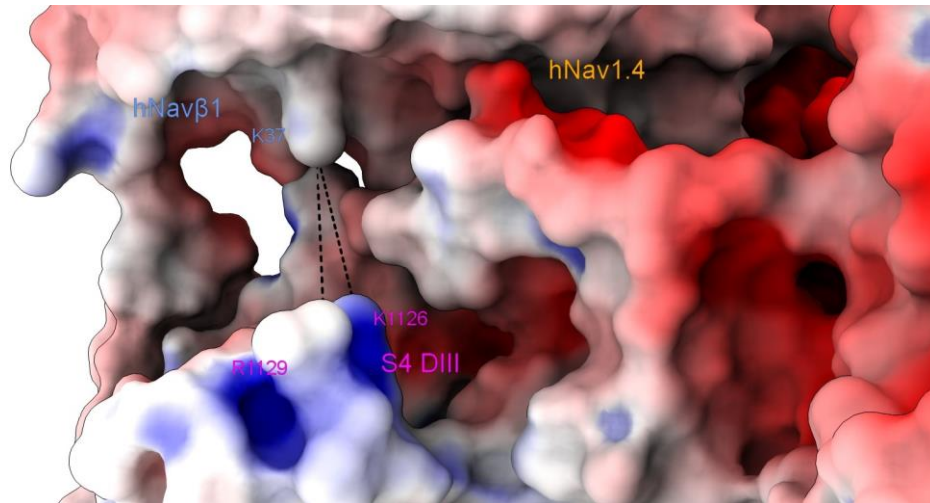
En esta investigación la hipótesis se deriva del desplazamiento del S4 DIII hacia la región extracelular durante la despolarización de la membrana celular cuando entra en contacto con Nav β 1 o Nav β 3, en donde las cargas de superficie en Nav α predomina la P-MEPS en el S4 DIII (por una lisina y una arginina en un segmento de secuencia altamente conservado gaiKsIRtlr) en Nav α están bajo la influencia de una posible colisión con una superficie electropositiva en Nav β 1 ó Nav β 3 por una lisina conservada en las tres especies (Tabla 5 y Figura 8).



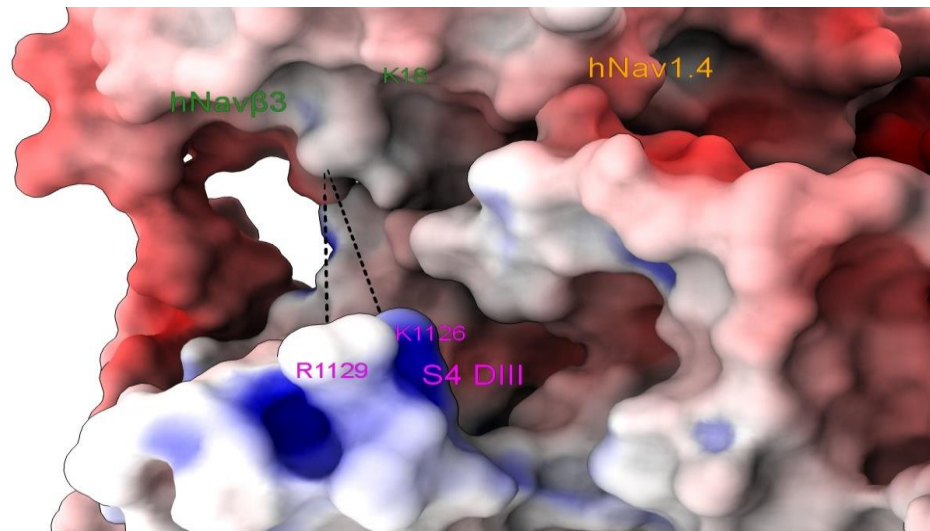
(a)



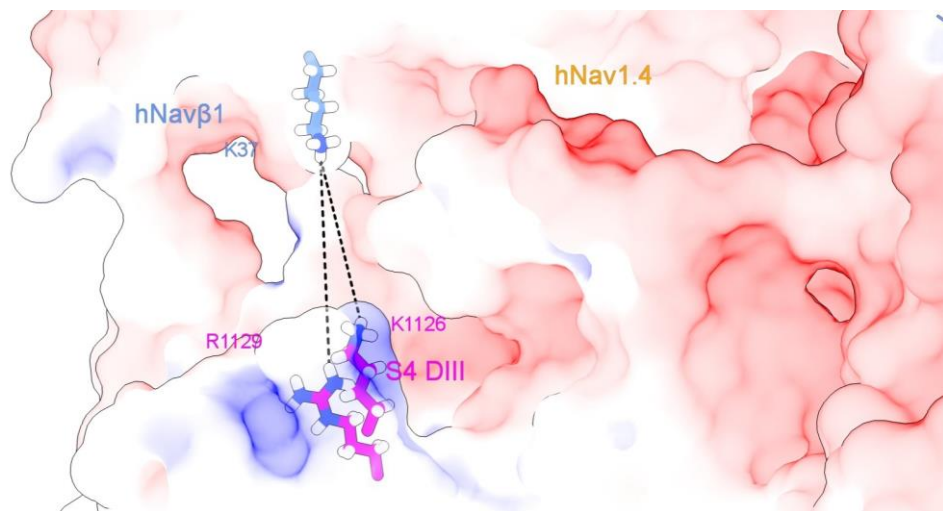
(b)



(c)



(d)



(e)

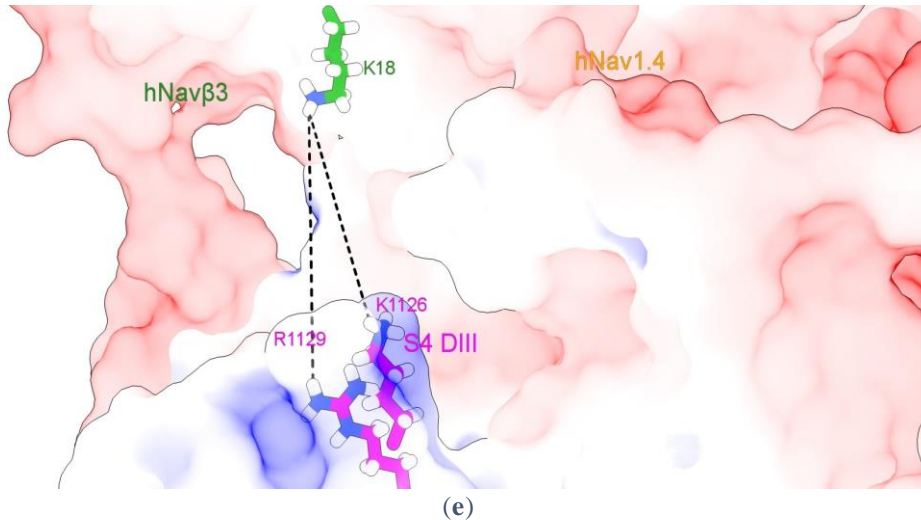


Figura 8. Sensor de voltaje S4 DIII de hNav1.4 α en estrecho contacto con hNav β 1 o hNav β 3. (a) hNav1.4 α interconectado con hNav β 1; (b) hNav1.4 α interconectado con Nav β 3; (c, y e) el MEPS en la interfaz hNav1.4 α /hNav β 1; (d, y f) el MEPS en la interfaz entre hNav1.4 α y hNav β 3. Las estructuras se prepararon con el complemento PDB2PQR [Dolinsky *et al.*, 2007] de Chimera y se calcularon las MEPS para superficies PPI utilizando Adaptive Poisson-Boltzmann Solver (APBS) [Baker *et al.*, 2001], una herramienta complementaria en Chimera Alpha 1.14 [Pettersen *et al.*, 2004] y simulada con Chimera X [Goddard *et al.*, 2017]. Sensor de voltaje S4 DIII: magenta; hNav β 1: azulcyan; hNav β 3: verde.

No es fuera de lo común suponer que, si un sensor de voltaje se acerca lo suficiente para la unión no covalente con ectodominios de IgD, se producirá una reordenación espacial.

En consecuencia, este desplazamiento podría desencadenar la modulación de la subunidad de los poros (Nav α). Además, suponemos que este contacto ejerce una rotación del IgD que afecta la inactivación rápida de Nav α gating.

Estos hallazgos indican que el S4 DIII podría crear una repulsión electrostática con las subunidades Nav β 1 o Nav β 3. Como consecuencia mecanicista sugerida, las IF-ECL mueven y tiran de la ECR, que a su vez transfiere energía de deformación a la subunidad Nav α . Esto ocurre precisamente en los S4 (DI y DIV) y podría generar un cambio conformacional en el canal que acelera la inactivación rápida y finalmente cierra el ciclo de modulación con S4 en DI y DIV volviendo a sus posiciones iniciales.

Los resultados quimiométricos y estructurales, consecuentemente conducen a la pregunta obvia que, si Nav β 1 o Nav β 3 tienen un posible contacto por repulsión de cargas con Nav α , sería posible que si Nav β 2 o Nav β 4 se unan a Nav α mediante el mismo mecanismo, y posiblemente tengan contacto con un sensor de voltaje hasta la fecha desconocido, en donde hipotéticamente puedan tener una influencia desconocida en la modulación de los Nav α que podría estar involucrada en el asa extracelular S5 DII (Pan *et al.*, 2019). Estas conjeturas de Nav β 2 o Nav β 4 no se discuten en esta investigación.

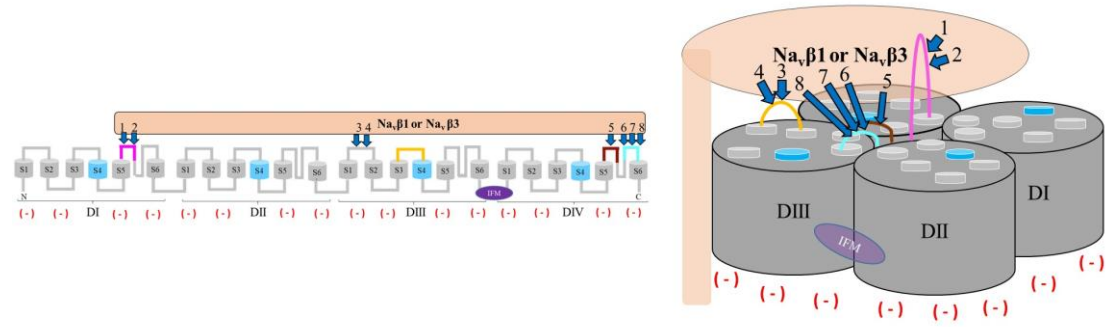
Tabla 5. Residuos de contacto del S4 DIII con las subunidades Nav β 1 y Nav β 3.

Isoforma	Na, α		Na, β					
	Código	¹ Residuos S4 DIII	Subunidad	Código	² Residuos	Subunidad	Código	² Residuos
hNav1.1 α	⁴ P35498	gai Ks Rtlr						
hNav1.2 α	³ 6J8E	gai Ks Rtlr						
hNav1.3 α	⁴ Q9NY46	gai Ks Rtlr						
hNav1.4 α	³ 6AGF	gpi Ks Rtlr						
hNav1.5 α	⁴ Q14524	gpi Ks Rtlr	hNa, β 1	³ 6AGF	mtf K ilc	hNa, β 3	³ 4L1D	npm K lrc
hNav1.6 α	⁴ Q9UQD0	gai Ks Rtlr						
hNav1.7 α	³ 6J8G	gpi Ks Rtlr						
hNav1.8 α	⁴ Q9Y5Y9	api Ka lRtlr						
hNav1.9 α	⁴ Q9UI33	me lK sfRtlr						
mNav1.1 α	⁴ A2APX8	gai Ks Rtlr						
mNav1.2 α	⁴ B1AWN6	gai Ks Rtlr						
mNav1.3 α	⁴ A2ASI5	gai Ks Rtlr						
mNav1.4 α	⁴ Q9ER60	gpi Ks Rtlr						
mNav1.5 α	⁴ Q9JIV9	gpi Ks Rtlr	mNa, β 1	⁴ P97952	mtf K ilc	mNa, β 3	⁴ Q8BHK2	nsm K lrc
mNav1.6 α	⁴ Q9WTU3	gai Ks Rtlr						
mNav1.7 α	⁴ Q62205	gpi Ks Rtlr						
mNav1.8 α	⁴ Q6QIY3	asi Ka lRtlr						
mNav1.9 α	⁴ Q9R053	pn lK sfRnlr						
rNav1.1 α	⁴ P04774	gai Ks Rtlr						
rNav1.2 α	⁴ P04775	gai Ks Rtlr						
rNav1.3 α	⁴ P08104	gai Ks Rtlr						
rNav1.4 α	⁴ P15390	gpi Ks Rtlr						
rNav1.5 α	⁴ P15389	gpi Ks Rtlr	rNa, β 1	⁴ Q00954	mtf K ilc	rNa, β 3	⁴ Q9JK00	npm K lrc
rNav1.6 α	⁴ O88420	gai Ks Rtlr						
rNav1.7 α	⁴ O08562	gpi Ks Rtlr						
rNav1.8 α	⁴ Q62968	asi Ka lRtlr						
rNav1.9 α	⁴ O88457	ps lK sfRtlr						

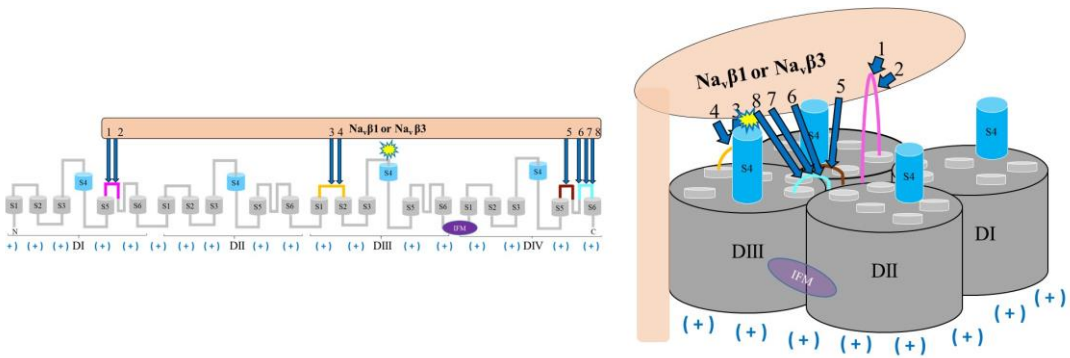
¹Residuos del S4 DIII que interactúan con las subunidades Nav β 1 y Nav β 3; ²Residuos de las subunidades Nav β 1 y Nav β 3 que interactúan con el sensor de voltaje S4 DIII; ³Código PDB: <http://www.rcsb.org/>; ⁴Código de Uniprot: <https://www.uniprot.org/>. Los aminoácidos que interactúan están en negrita y letras mayúsculas, e. g. **K** y **R**.

En base en los presentes hallazgos, proponemos un posible mecanismo de modulación de la subunidad α formadora del poro por subunidades β para la aceleración de la

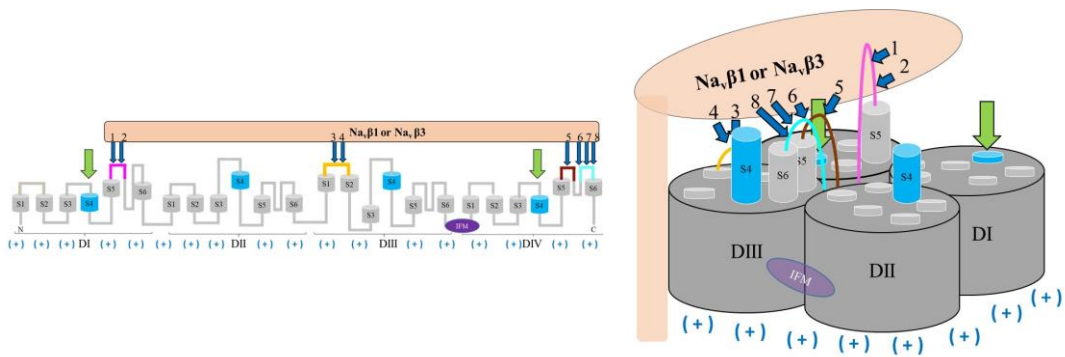
inactivación rápida (Figura 9). En los modelos PPI, se aplicó una biblioteca de rotámeros para encontrar contactos favorables de van der Waals [Pettersen *et al.*, 2008, Scouras *et al.*, 2011, Shapovalov *et al.*, 2011] en relación a Nav α /Nav β 1 y Nav α /Nav β 3 (Tabla S1).



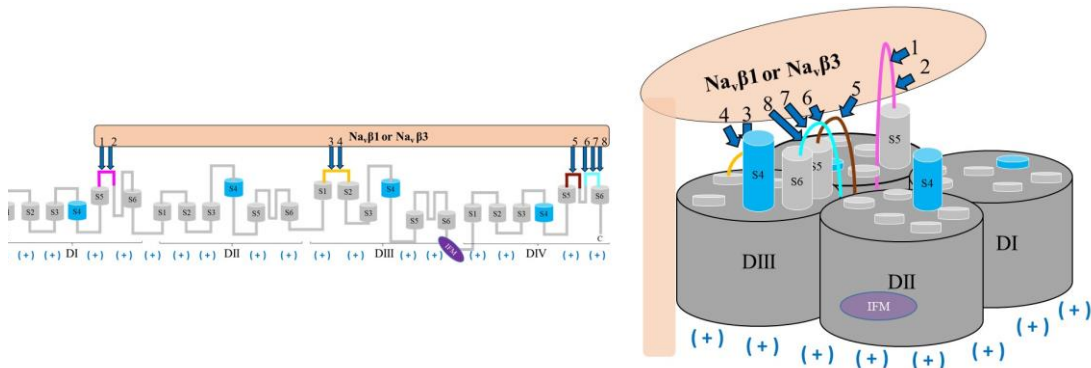
(a)



(b)



(c)



(d)

Figura 9. Modulación hipotética de la inactivación rápida del gating Nav_α por $Nav\beta 1$ o $Nav\beta 3$.

(a) Nav_α en estado inactivo (cerrado) en interacción con $Nav\beta 1$ o $Nav\beta 3$; (b) Nav_α en estado abierto (activado) en presencia de modulación $Nav\beta 1$ o $Nav\beta 3$; (c) inactivación rápida modulada por $Nav\beta 1$ o $Nav\beta 3$; (d) inactivación rápida desencadenada por la puerta de inactivación de IFM; números 1 a 8: PPI-Ids de las PPI polares calculadas (Tabla S1); PPI polares calculados: flechas azul marino; el retorno a su posición inicial de S4 forzado por las IF-ECL mediante interacciones polares calculadas con $Nav\beta 1$ o $Nav\beta 3$: flechas verdes; cargas negativas: signos negativos rojos entre paréntesis; cargas positivas: signos positivos azul marino entre paréntesis; regiones Nav_α sin PPI: gris claro y oscuro; subunidad $Nav\beta 1$ o $Nav\beta 3$: salmón claro; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cyan; S4: azul cielo, segmentos S1, S2, S3, S4, S5 y S6: gris claro.

8.3. Determinación de las propiedades de las ECLs

Se cuantificaron todos los tamaños de las ECLs de las nueve isoformas para las tres especies las cuales se muestran en la Figura 10.

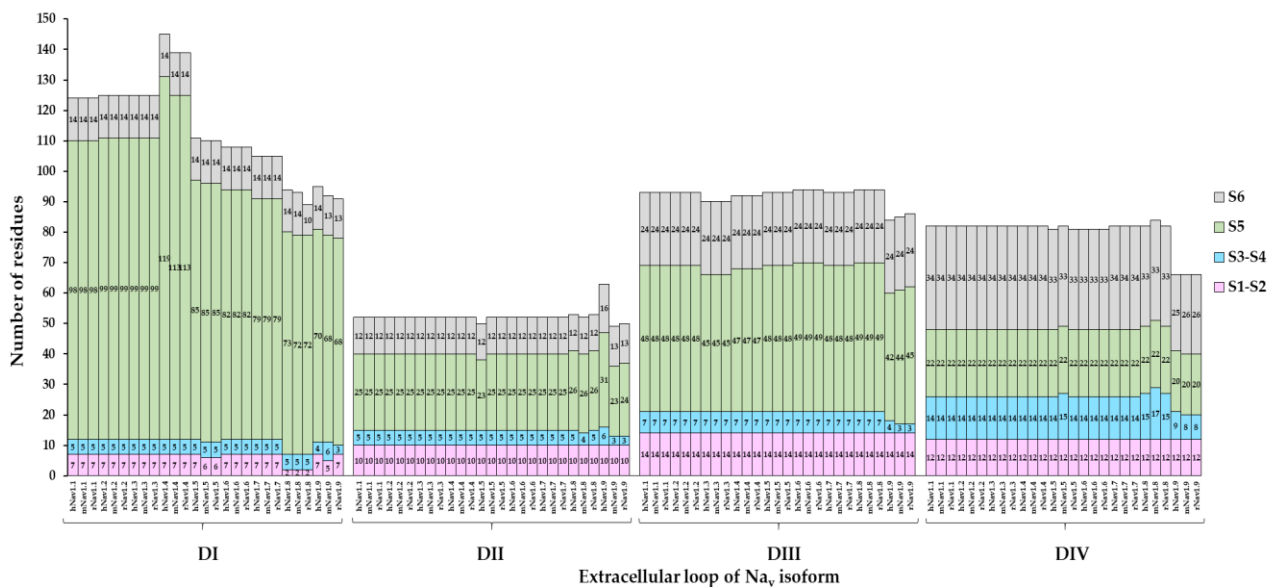


Figura 10. Longitud de las ECLs de las Navas.

ECLs: S1-S2, S3-S4, S5 y S6.

Además, se determinaron las propiedades de la ECR y las ECLs S5 y S6, las cuales se presentan en la Figura 11.

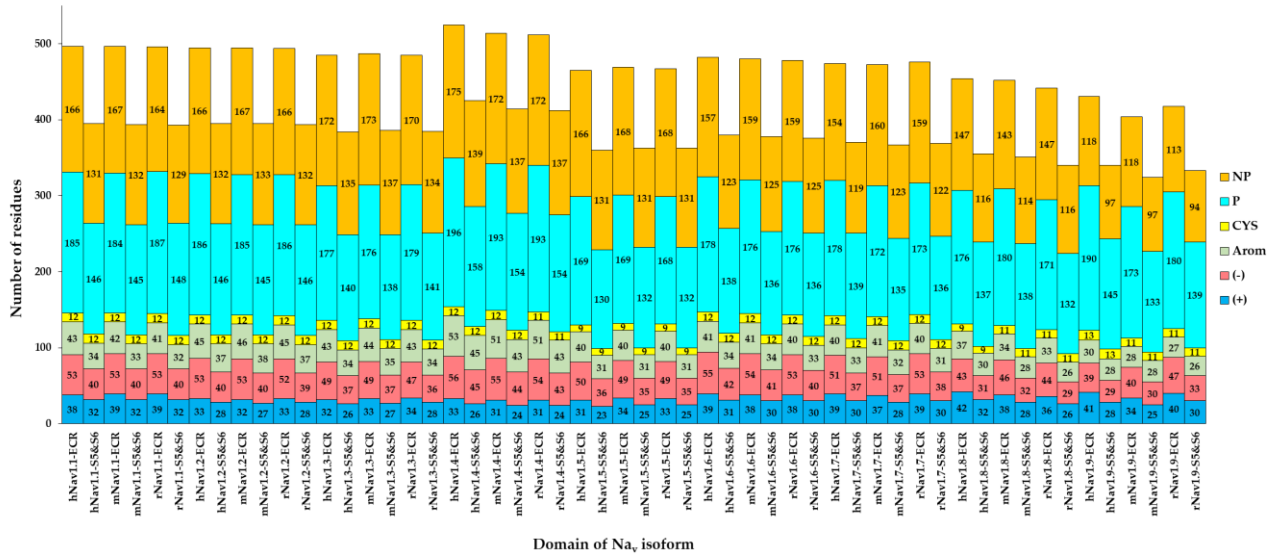
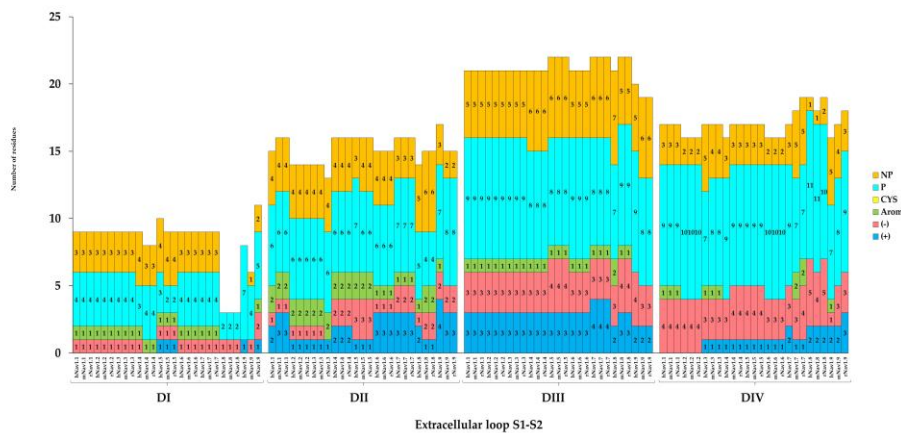
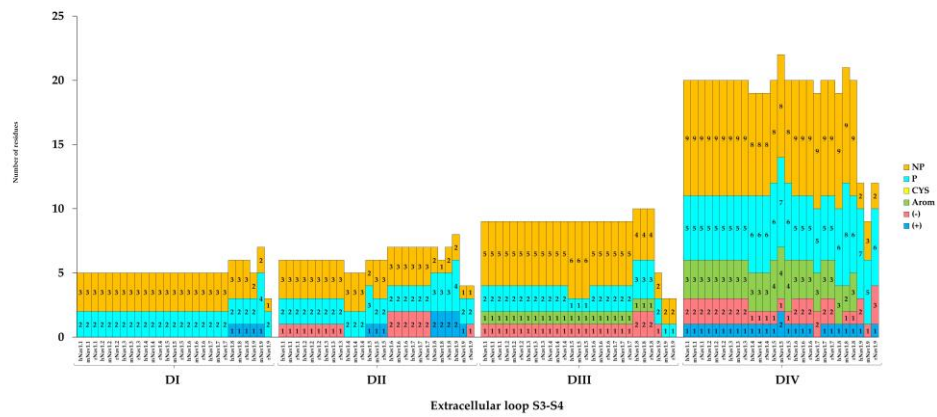


Figura 11. Propiedades de los residuos de la ECR de las Navas. S5 y S6: representan las asas extracelulares por isoforma; NP: no polar; P: polar; CYS: cisteínas; Arom: residuos aromáticos; (-): Cargado negativamente; (+): cargado positivamente. Los residuos polares y apolares representan los residuos totales de la región analizada.

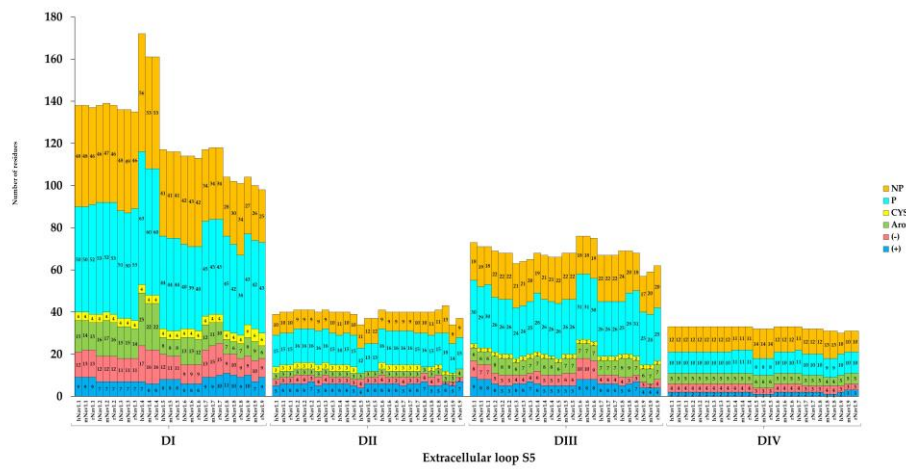
También, se determinaron las propiedades de los residuos de las ECLs de la subunidad α , Figura 12.



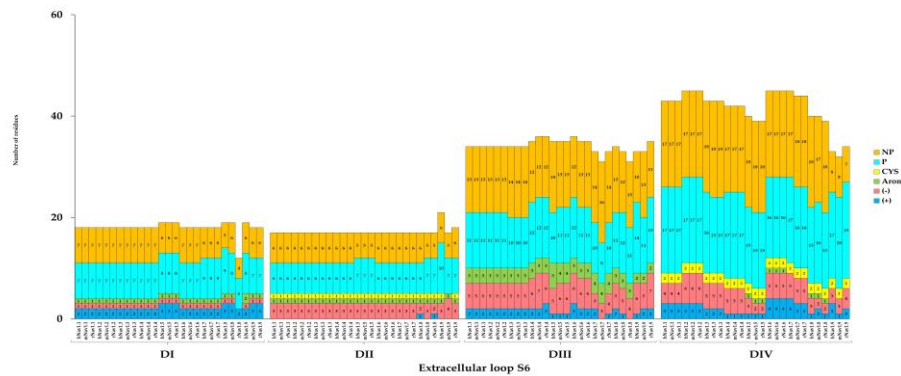
(a)



(b)



(c)



(d)

Figura 12. Propiedades de los residuos de las ECLs de las Navas.

(a) ECL S1-S2; (b) ECL S3-S4; (c) ECL S5; (d) ECL S6; NP: no polar, P: polar, CYS: cisteínas; Arom: residuos aromáticos; (-): residuos cargados negativamente; (+): residuos cargados positivamente. Los residuos polares y a polares representan los residuos totales de la región analizada.

Los tamaños de las ECLs, las propiedades de la ECR y las ECLs S5 y S6 y las propiedades de los residuos de las ECLs de las Navas, se determinaron mediante ecuaciones empíricas en [Microsoft Excel, 2016]. Cada isoforma y ECL presenta características inherentes específicas, que se discuten con más detalle en discusión de resultados.

Debido a las características inherentes de cada ECR, es de suponer que cada isoforma formadora de poro varía en cuanto a superficie y volumen.

Como un parámetro adicional para calcular variaciones en el ECR entre isoformas y especies, se calculó el volumen molecular y la SAA de la ECRs y las ECLs S5 y S6, las cuales se muestran en la Figura 13.

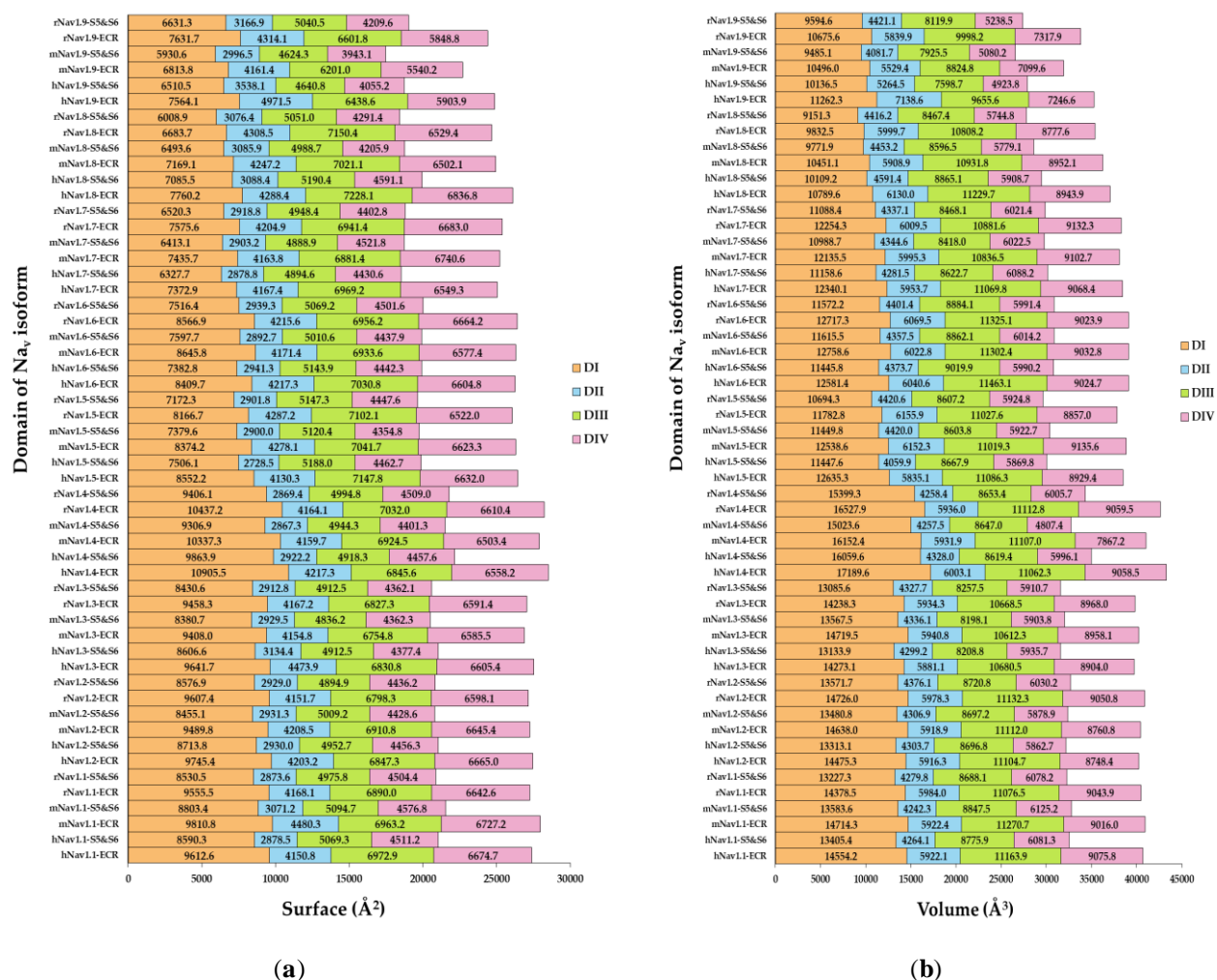


Figura 13. Propiedades de SAA y volumen de la ECR y ECLs S5 y S6 de las Navas. (a) SAA y (b) volumen molecular; S5 y S6: asas extracelulares S5 y S6; DI - DIV: dominios I - IV, respectivamente.

De manera divergente a los resultados de la Figura 13, se calculó el volumen molecular y la SAA de las ECLs, las cuales se presentan en las Figuras 14 y 15.

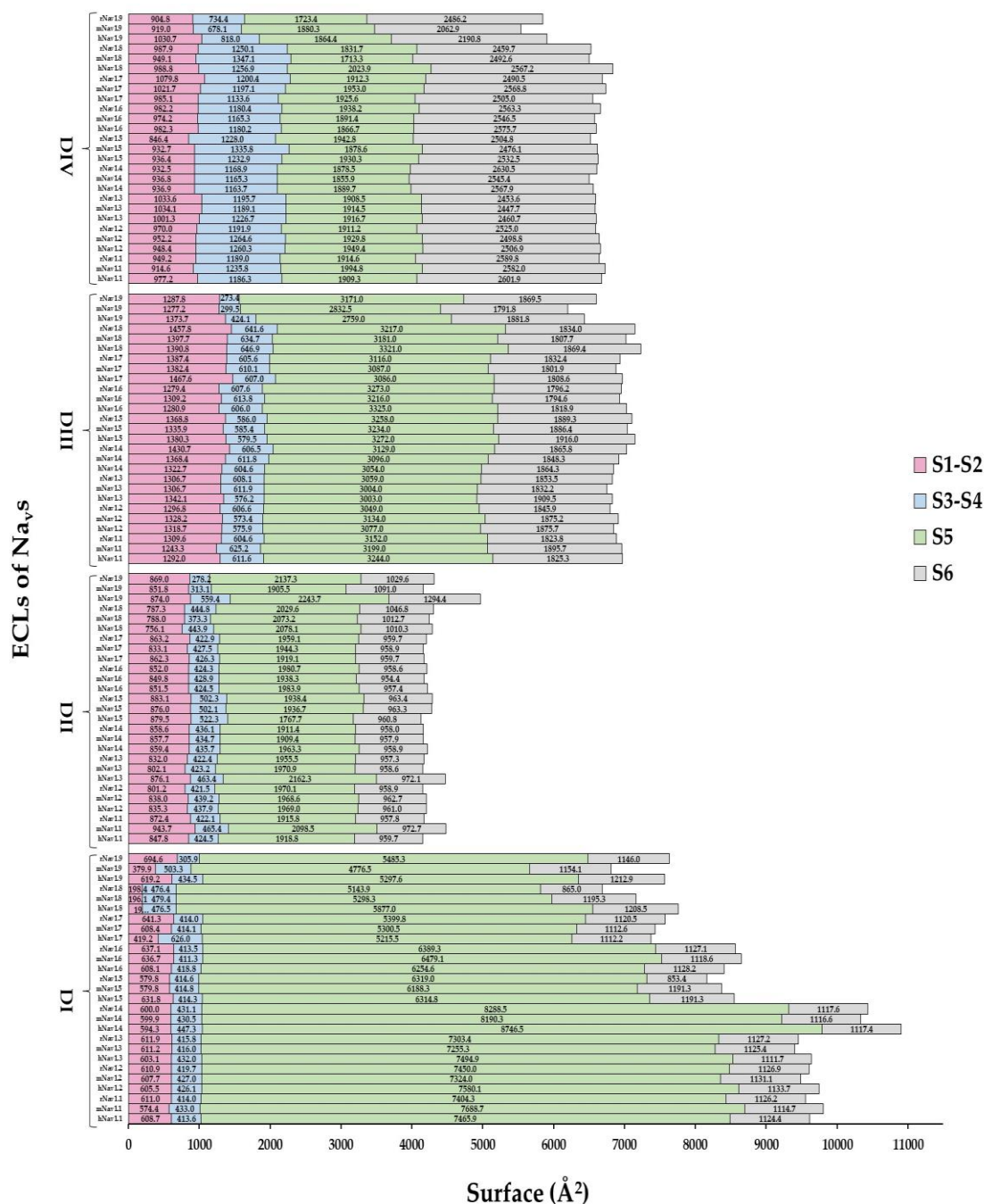


Figura 14. SAA de las ECLs de las Navas.
DI – DIV: dominios I – IV, respectivamente.

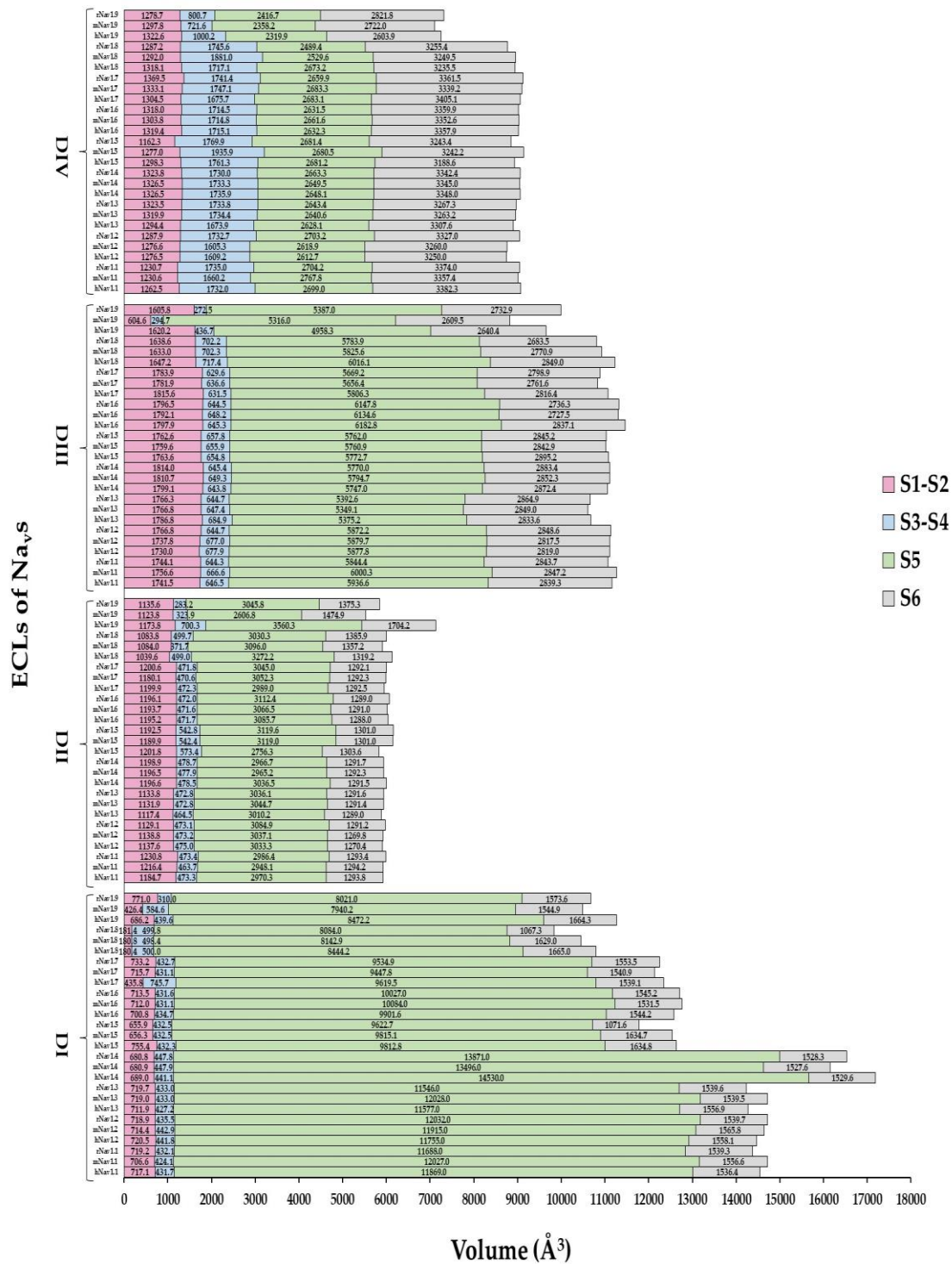
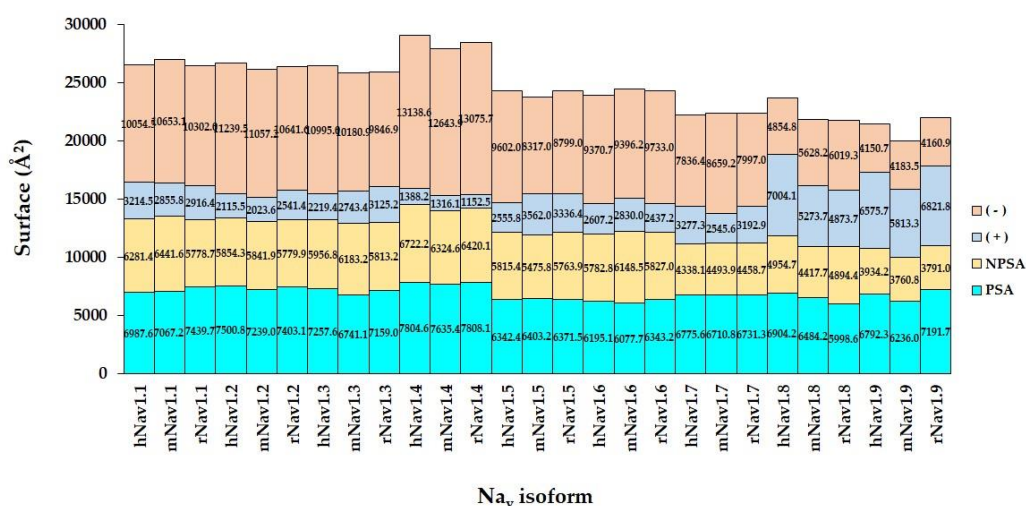
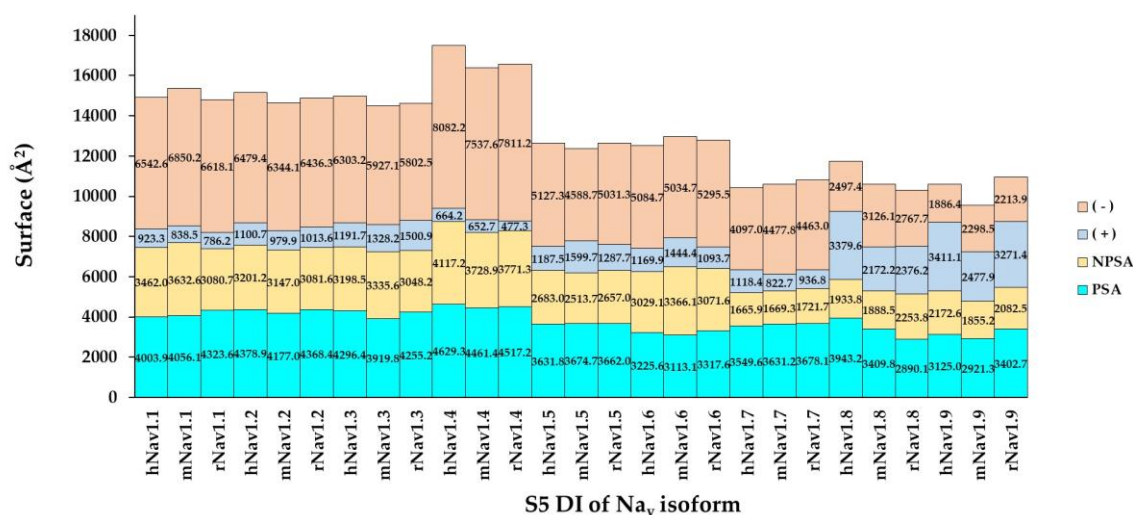


Figura 15. Volumen molecular de las ECLs de las Navas. DI – DIV: dominios I – IV, respectivamente.

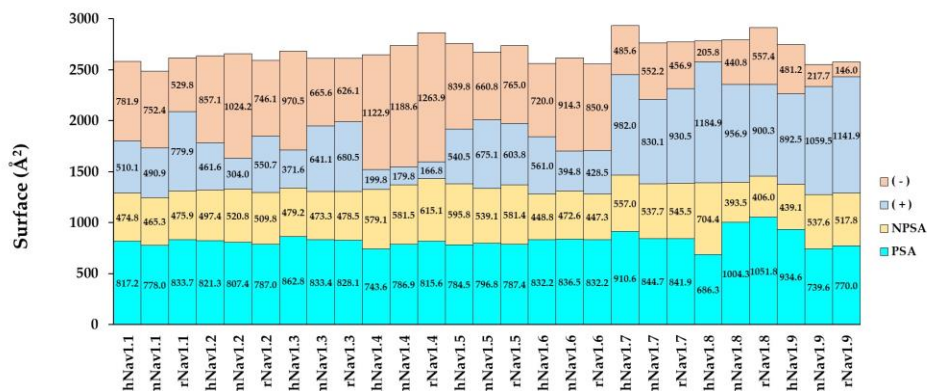
Debido al tamaño, composición de residuos y estructura 3D variable de los modelos de las IF-ECLs y la ECR, es de suponer que presentan propiedades de superficie variables con características inherentes que son puntos clave en la interfaz de las IF-ECLs. El área de superficie polar (PSA), el área de superficie no polar (NPSA) y MEPS (potencial electrostático de superficie) negativo: N-MEPS y MEPS positivo (P-MEPS) se calcularon para todas las IF-ECL como un total general y valores individuales para cada IF-ECL (Figura 16).



(a)

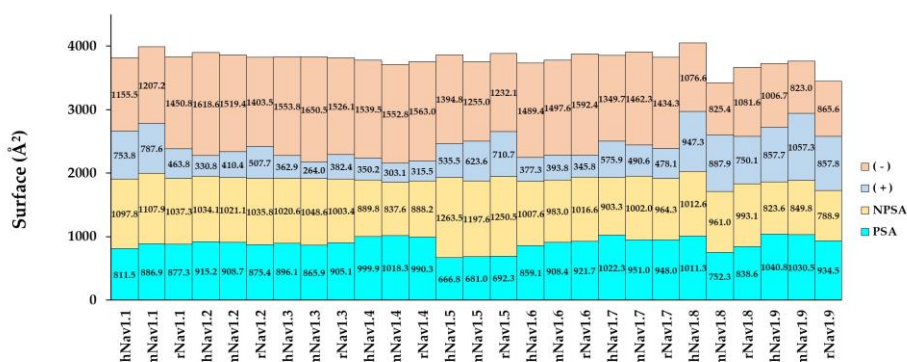


(b)



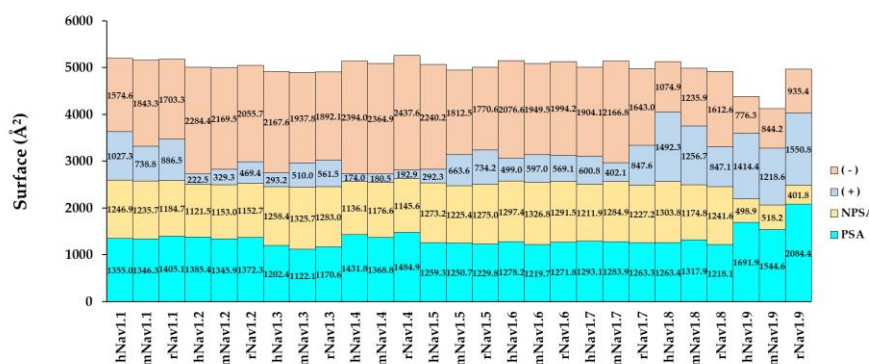
S1-S2 DIII of N_{av} isoform

(c)



S5 DIV of N_{av} isoform

(d)



S6 DIV of N_{av} isoform

(e)

Figura 16. Propiedades de SAA de la ECR y las IF-ECL de las Navas.

(a) ECRs las Navas; (b) S5 DI; (c) S1-S2 DIII (d) S5 DIV; (e) S6 DIV; PSA: superficie polar; NPSA: superficie no polar; (+): P-MEPS; (-): N-MEPS. El PSA y NPSA representan el área total de la región analizada.

8.4. Propiedades de interfaz y Patrones de interacción en los Navs

La Tabla S2, muestra el área enterrada de interfaz que se forma entre las IF-ECLs y las Nav β s. Como parámetro adicional, se determinó el área SAS de los átomos que forman el área enterrada entre α y β , es decir, el area SAS completa de los átomos que forman la intersección espacial de las IF-ECLs con Nav β 1 y Nav β 3 (Figura 17).

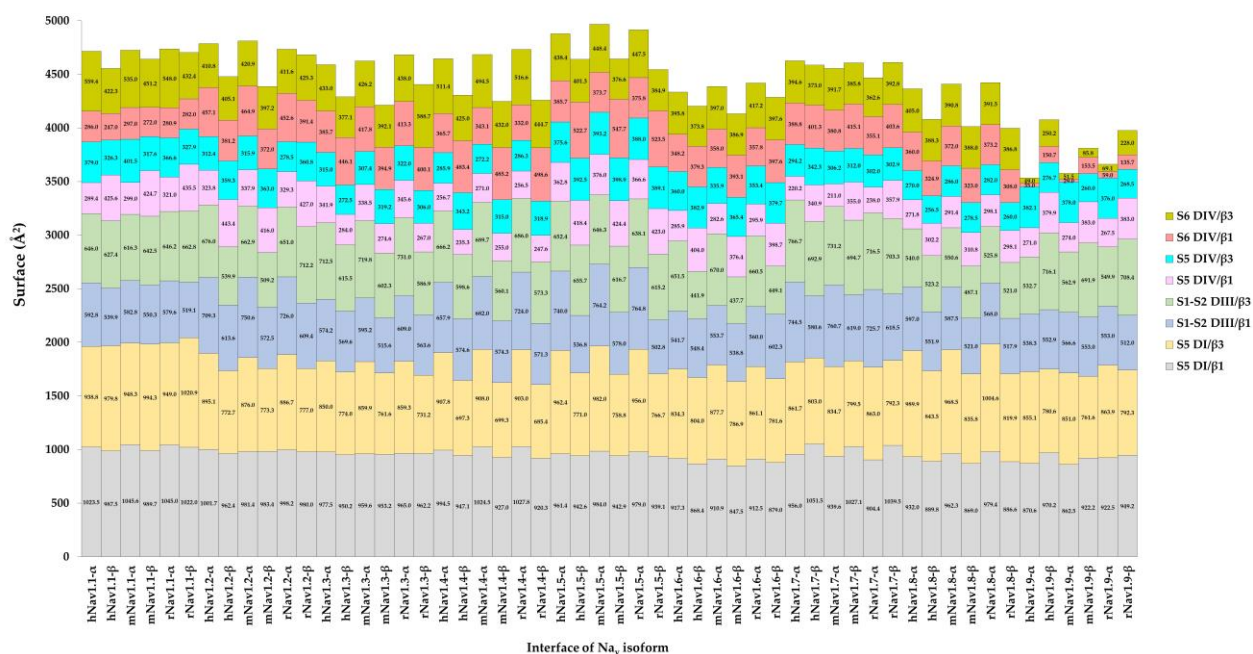


Figura 17. Área total de los átomos que forman la interfaz Nava/Nav β . α : área de los átomos de Nava que forma la interfaz; β : área de los átomos de Nav β que forma la interfaz.

Para evitar el sesgo de la longitud del asa (las ECLs más largas tienden a tener más posibilidades de formar una mayor área de interfaz enterrada que las ECLs más cortas), se decidió presentar valores normalizados, como regla general: se toma el porcentaje de IF del total mencionado anteriormente como base del 100%. Se calcularon las siguientes propiedades en la interfaz para las IF-ECLs con ambas subunidades β : PSA, NPSA, P-MEPS y N-MEPS (Figuras 18 a 21).

En el siguiente paso, se detectaron características comunes en las propiedades de interfaz y se agruparon en los siguientes patrones de interacción: S5 DI: (Nav1.1 α y Nav1.3 α), (Nav1.5 α y Nav1.7 α), (Nav1.2 α , Nav1.4 α y Nav1.6 α), (Nav1.8 α) y (Nav1.9 α); S1-S2 DIII: (Nav1.1 α y Nav1.4 α), (Nav1.3 α y Nav1.5 α), (Nav1.2 α y Nav1.6 α), (Nav1.8 α) y (Nav1.9 α); S5 DIV: (Nav1.1 α y Nav1.5 α), (Nav1.2 α , Nav1.3 α y Nav1.6 α), (Nav1.4 α y

Nav1.7 α), (Nav1.8 α) y (Nav1.9 α). En el caso del S6 DIV, no se encontraron similitudes entre las nueve isoformas.

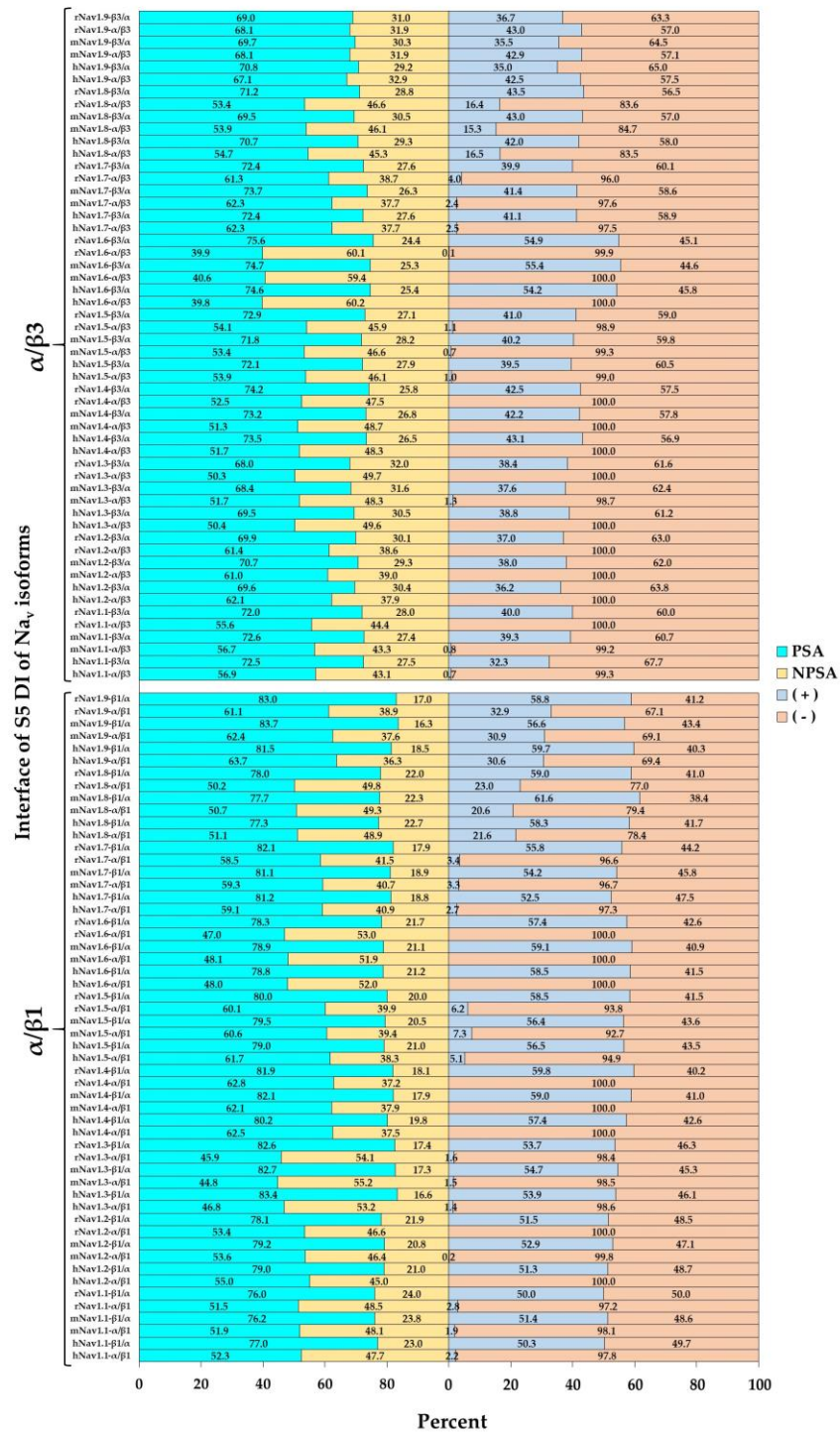


Figura 18. Porcentajes de las propiedades de superficie de interfaz de Nav β 1 y Nav β 3 en complejo con el S5 DI. α/β : superficie de la subunidad α que participan en la interfaz; β/α : superficie de la subunidad β que participan en la interfaz; PSA: superficie polar; NPSA: superficie no polar; (+): P-MEPS; (-): N-MEPS. El PSA y NPSA representa el área total de la interfaz analizada.

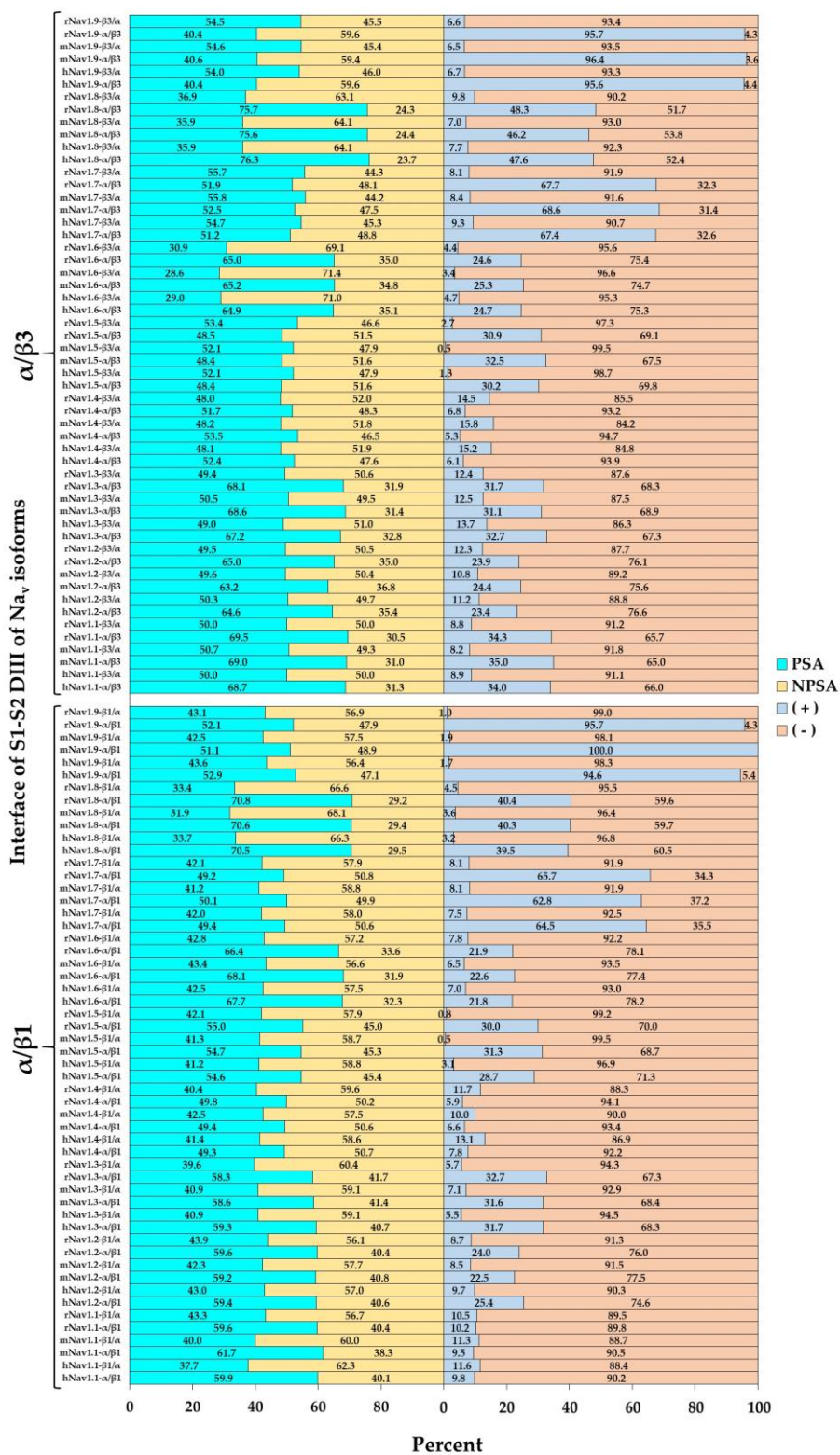


Figura 19. Porcentajes de las propiedades de superficie de interfaz de Nav β_1 y Nav β_3 en complejo con el S1-S2 DIII.

α/β : superficie de la subunidad α que participan en la interfaz; β/α : superficie de la subunidad β que participan en la interfaz; PSA: superficie polar; NPSA: superficie no polar; (+): P-MEPS; (-): N-MEPS. El PSA y NPSA representa el área total de la interfaz analizada.

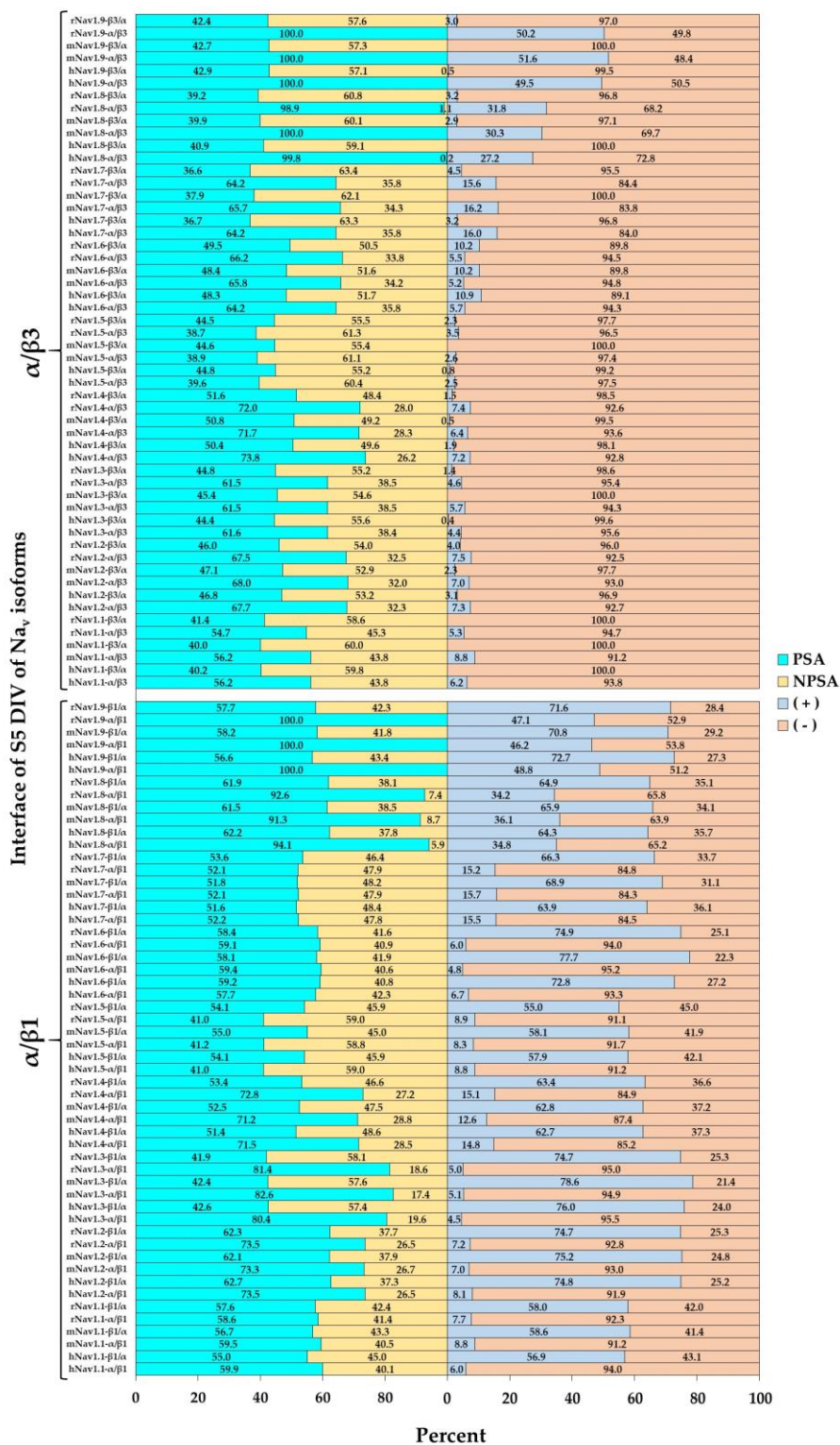


Figura 20. Porcentajes de las propiedades de superficie de interfaz de Navβ1 y Navβ3 en complejo con el S5 DIV. α/β: superficie de la subunidad α que participan en la interfaz; β/α: superficie de la subunidad β que participan en la interfaz; PSA: superficie polar; NPSA: superficie no polar; (+): P-MEPS; (-): N-MEPS. El PSA y NPSA representa el área total de la interfaz analizada.

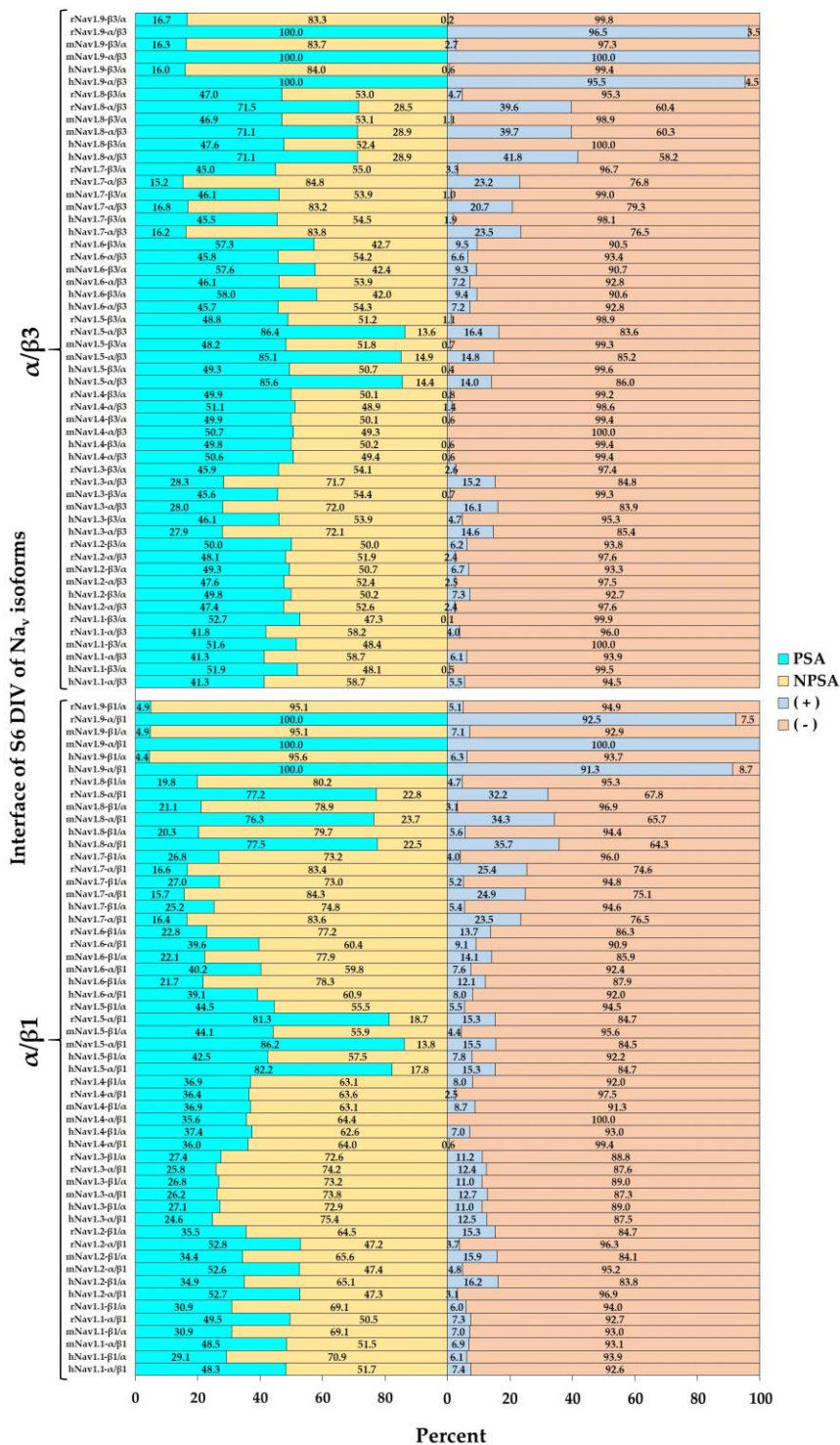
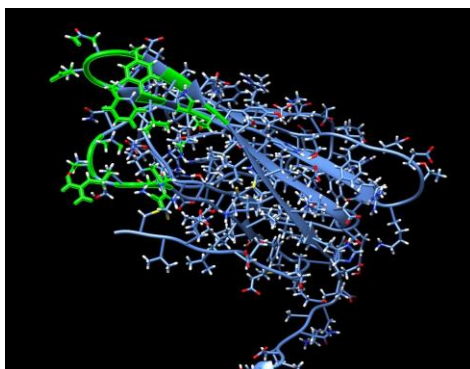
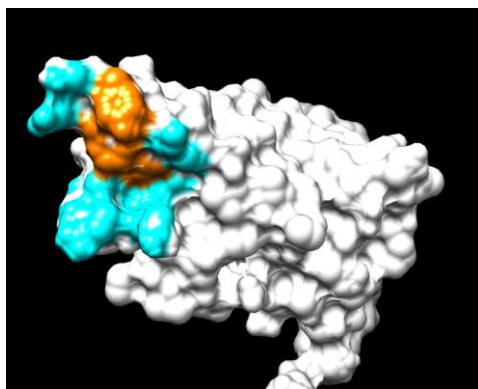


Figura 21. Porcentajes de las propiedades de superficie de interfaz de Navβ1 y Navβ3 en complejo con el S6 DIV. αβ: superficie de la subunidad α que participan en la interfaz; βα: superficie de la subunidad β que participan en la interfaz; PSA: superficie polar; NPSA: superficie no polar; (+): P-MEPS; (-): N-MEPS. El PSA y NPSA representa el área total de la interfaz analizada.

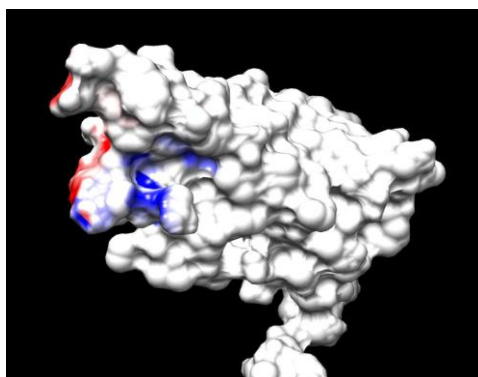
La Figura 22, ilustra como ejemplo demostrativo, la superficie de interfaz de la subunidad $\beta 1$ y sus propiedades en todos los detalles.



(a)



(b)



(c)

Figura 22. Ejemplo demostrativo de las propiedades de superficie de los átomos que forman la interfaz en $hNav\beta 1/S5$ DI del complejo $hNav\beta 1/hNav1.4\alpha$.

(a) átomos que forman la interfaz en verde; (b) área de interfaz: PSA y NPSA en cian y naranja, respectivamente; (c) área de interfaz: P-MEPS y N-MEPS en azul y rojo, respectivamente; superficie que no forma interfaz en blanco. La MEPS se calculó para las estructuras y modelos individuales utilizando las herramientas de complemento PDB2PQR y Adaptive Poisson-Boltzmann Solver (APBS) [Baker *et al.*, 2001; Dolinsky *et al.*, 2007] en Chimera Alpha 1.14 [Pettersen *et al.*, 2004]. Es de destacar que las áreas de átomos que contribuyen a la interfaz no constituyen necesariamente áreas enterradas (cf. Tabla S2).

9. DISCUSIÓN DE RESULTADOS

9.1. Patrones de PPI en las isoformas de los Navs

El fundamento de las funciones de las proteínas en la mayoría de los procesos biológicos constituye la plétora de interacciones átomo-átomo entre proteínas y otras biomoléculas (cf. interactoma). La predicción de interacciones a nivel atómico sigue siendo uno de los esfuerzos más desafiantes en biología estructural [Hermann *et al.*, 2007; Zahiri *et al.*, 2013; Ezkurdia *et al.*, 2008].

Durante la evolución, la estructura de las proteínas se conserva más que su secuencia primaria subyacente, es decir, las secuencias divergen de un ancestro común, pero mantienen funciones idénticas o similares con pequeños cambios debido a intercambios homólogos en sus sitios activos [Whisstock *et al.*, 2003; Izidoro *et al.*, 2014; Jacobson *et al.*, 2014]. Los residuos en la sensibilidad de las interfaces se vuelven importantes para la geometría o la señalización y, por lo tanto, tienden a conservarse en la estructura de la proteína [Ma *et al.*, 2003]. Otra propiedad bien caracterizada de las interfaces se refiere a la existencia de residuos importantes de interacción "hot spots", que son los residuos que hacen las mayores contribuciones a la formación de complejos [Keskin *et al.*, 2005]. En el contexto de nuestro estudio, varios informes han planteado la cuestión de qué componentes atómicos están hechas exactamente las interfaces de proteínas para mejorar el poder de predicción de las PPI [Li *et al.*, 2004].

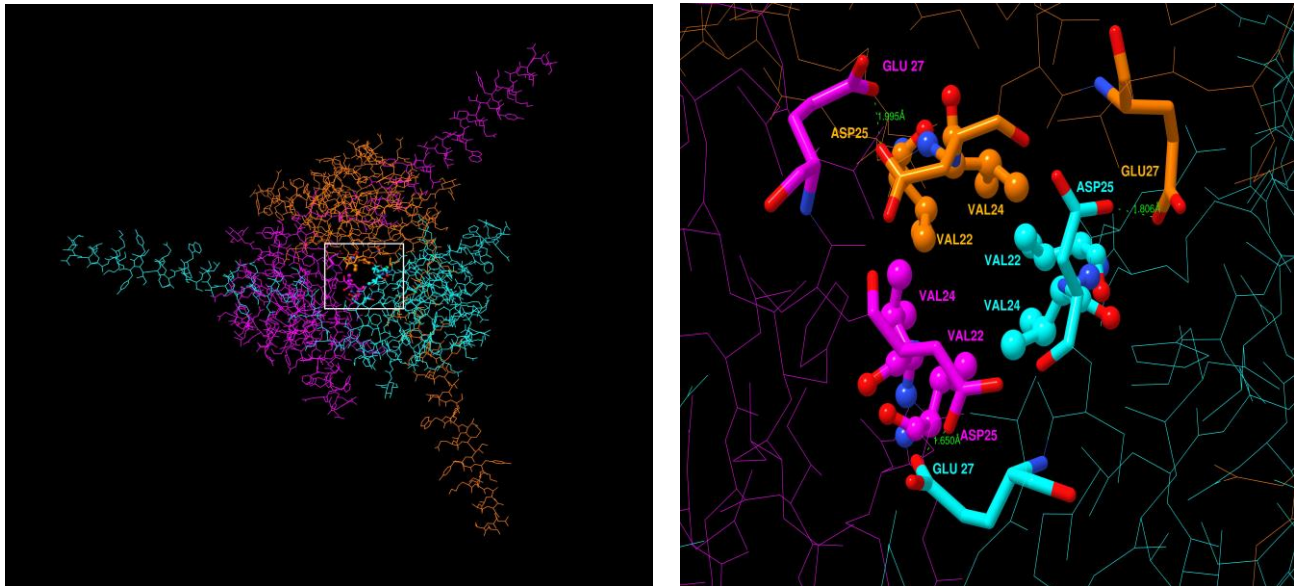
La investigación *in vitro* para obtener información sobre los mecanismos de los Navs a escala atómica ha sido una tarea abrumadora durante décadas debido a la ubicación de la subunidad Nav α formadora del poro integrada en la membrana lipídica y su complejidad multidominio [Catterall, 2000; Kruger *et al.*, 2006]. Es en tales situaciones cuando los enfoques quimiométricos brindan una visión que desentraña patrones atómicos que subyacen en funciones biológicas, que hasta la fecha han pasado desapercibidas.

Las subunidades de Nav β pertenecen a la superfamilia de inmunoglobulinas (Ig). Su estructura general es un pliegue de láminas beta típicas de las moléculas de adhesión celular [Salvage *et al.*, 2020]. La variación en presencia (o ausencia) de subunidades β regula la expresión de la subunidad α para diferenciar tejidos. Además, modulan la cinética de la unidad formadora de poro [Calhoun e Isom; 2014; Isom *et al.*, 1995], en donde el mecanismo por el cual se produce este fenómeno se ha estudiado ampliamente y aún permanece incierto.

Sin embargo, en esta investigación, un indicio importante de manera hipotética de la comprensión del mecanismo de la modulación de la aceleración de la inactivación rápida del canal y su modulación, en detalle acaba de emerger de una manera bit a bit.

Echando un vistazo hacia atrás en la historia de la investigación de los Navs, en los años 1985 y 2000 las subunidades Nav β 1 y Nav β 3, respectivamente fueron reportadas por primera vez como proteínas de adhesión celular (de superficie) en interacción con la unidad α formadora de poro a través de enlaces no covalentes [Messner y Catterall., 1985; Morgan *et al.*, 2000].

Nuevos aspectos estructurales de las subunidades Nav β , provienen del laboratorio de A.P. Jackson [Namadurai *et al.*, 2014] que han dilucidado una estructura cristalina trimérica de Nav β 3, es decir, tres IgD en una celda unitaria cristalográfica. No obstante, ha sido una pregunta sin resolver si el trímero también refleja una unidad biológica funcional. Recientemente, estudios *in silico* de dinámicas moleculares indicaron que la oligomerización espontánea de las subunidades de Nav β 3 de longitud completa en un trímero probablemente sería un proceso muy lento si ocurriera en las membranas celulares. Incluso si las tres TMH de Nav β 3 interactuaran, sería con fuerzas moleculares débiles [Glass *et al.*, 2020]. En este estudio, se realizó un análisis estructural para concretar si las IgD de las subunidades hNav β 1 podrían formar trímeros [Pan *et al.*, 2018; Namadurai *et al.*, 2014]. Determinamos que en las tres IgD de hNav β 1 existe una fuerte repulsión entre las cargas totales negativas de los residuos de Asp25 y Glu27 al alinearlos en las posiciones espaciales de cada monómero de la estructura trimérica Nav β 3 (PDB: 4L1D [Namadurai *et al.*, 2014]). Hasta ahora, este hallazgo no se ha reportado. Ambos aminoácidos permanecen conservados durante la evaluación a través de especies de mamíferos, lo que sugiere un punto clave de PPI fundamentales para las Nav α (Figura 23, Tabla S1 y Figuras S1 a S9). Glass *et al.*, 2020, razonaron que, si el trímero de hNav β 3 interactuara estructuralmente con los VSD de la proteína α formadora de poro en analogía con hNav β 1 conocida, sería necesaria una reordenación sustancial de las IgD de de hNav β 3.



(a)

(b)

Species

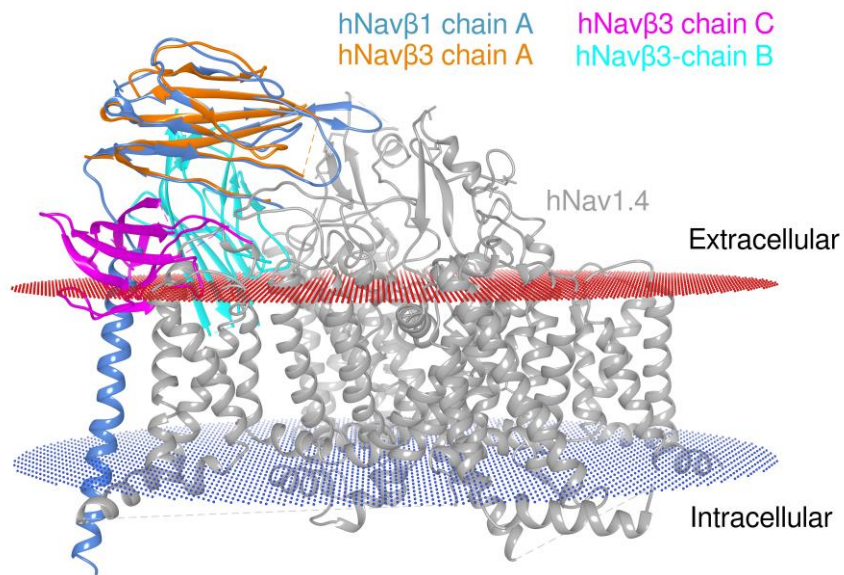
Homo sapiens
Mus musculus
Rattus norvegicus
Bos Taurus
Pan troglodytes
Canis lupus familiaris
Oryctolagus cuniculus
Equus caballus

GCVEVDSETEAV
GCVEVDS DTEAV
GCVEVDSETEAV
GCVEVDSETEAV
GCVEVDSETEAV
GCVEVDSETEAV
GCVEVDSETEAV
GCVEVDSETEAV
GCVEVDSETEAV

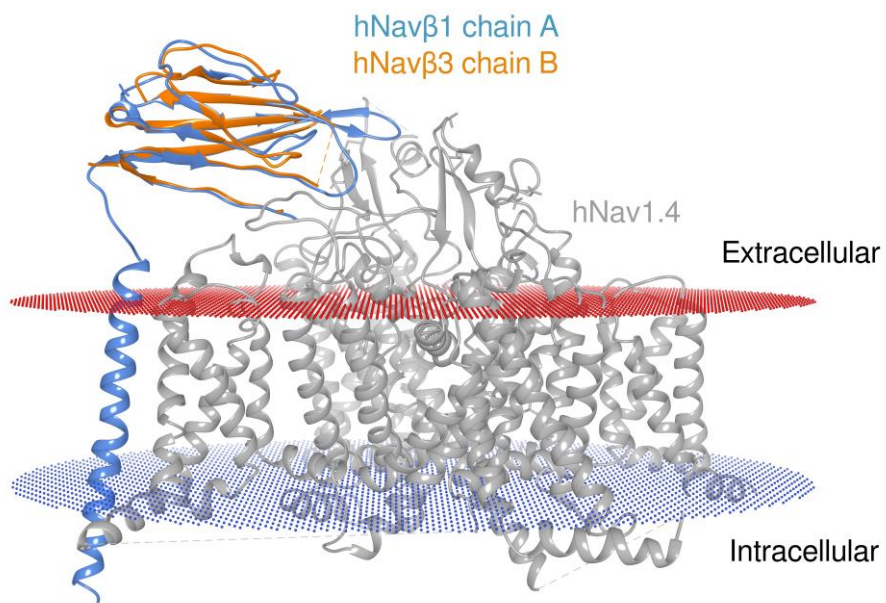
(c)

Figura 23. Presentación de un modelo de hNavβ1 trimérico. (a) Trímero de Navβ1 visto de arriba (lgD) hacia abajo (b) Vista de cerca desde arriba del supuesto hotspot (c) Secuencias alineadas de especies de mamíferos; cuadro blanco: área de análisis; líneas de puntos verdes: repulsión de cargas calculada; estructura en forma de bastones: residuos con carga negativa; estructura en forma de bastones y esferas: residuos que forman un posible núcleo hidrofóbico. Datos generados por Chimera Alpha 1.14 [Pettersen et al., 2004].

Dado que las interfaces de las estructuras determinadas por cristalografía de Navα/Navβ1 en el complejo existen además de Navβ3 trimérica y el Navβ1 monomérica (hNavβ3 [Namadurai *et al.*, 2014], hNavβ1 [Pan *et al.*, 2018]), fue posible realizar estudios de biología estructural superponiéndolos entre sí (Figura 24).



(a)



(b)

Figura 24. Superposición de proteínas Nav β monoméricas y triméricas en complejo con Nava.

(a) hNav β 3 trimérico determinado experimentalmente [Namadurai *et al.*, 2014] (tres colores: magenta, cyan, naranja) en superposición con la estructura molde hNav1.4 α /hNav β 1 (azul oscuro/gris) [Pan *et al.*, 2018] y (b) monomero IgD (de tres) de hNav β 3 homotrimérico (naranja) [Namadurai *et al.*, 2014] en superposición con la estructura molde hNav1.4 α /hNav β 1 [Pan *et al.*, 2018]. Se puede ver a simple vista, que dos de las tres subunidades choquen contra la membrana y la subunidad formadora del poro. Límites de la membrana extracelular: rojo oscuro; límites de la membrana intracelular: azul marino; subunidad Nava: gris; subunidad hNav β 1: azul oscuro; cadenas A, B y C de hNav β 3: naranja, cyan y magenta, respectivamente.

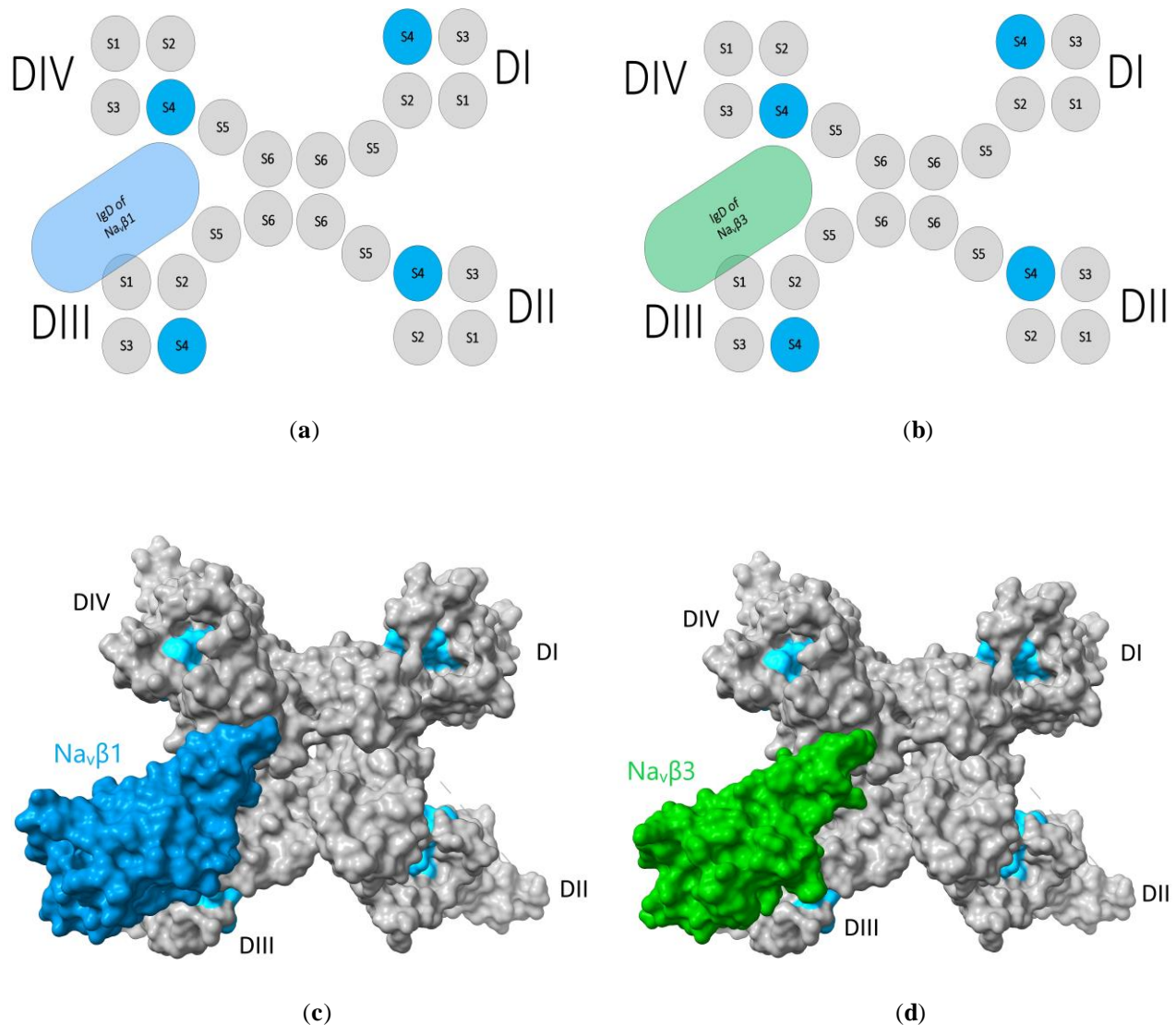


Figura 25. Ubicación tridimensional (3D) de las subunidades $Nav\beta 1$ y $Nav\beta 3$ en la subunidad formadora del poro $Nav\alpha$. (a, b) Topología de $Nav\alpha$ en complejo con $Nav\beta 1$ y $Nav\beta 3$. Visualización de estructuras 3D con superficies excluidas al disolvente en los paneles (c) y (d); (c) Estructura Cryo-EM de $hNav1.4\alpha$ en complejo con $Nav\beta 1$ [Pan *et al.*, 2018] y (d) Complejo estructural de Cryo-EM de $hNav1.4\alpha$ [Pan *et al.*, 2018] ensamblada estructuralmente (3D) con la estructura de cristal $hNav\beta 3$ [Namadurai *et al.*, 2014] posicionada de acuerdo al análisis estructural realizado; S4: azul cielo; en los paneles (c) y (d) las superficies moleculares excluidas al disolvente están coloreadas: $Nav\beta 1$: azul cian; $Nav\beta 3$: verde bosque; y color gris para las superficies de la subunidad $Nav\alpha$. Se aplicaron los mismos colores a los paneles (a) y (b) anteriores. Modelos 3D de Chimera X [Goddard *et al.*, 2017].

En comparación con el gran número de variaciones de todos los niveles β , también conocidos como miembros de la superfamilia de las Ig, el ectodominio IgD de $hNav\beta 3$ [Namadurai *et al.*, 2014] es extremadamente similar a la IgD de $hNav\beta 1$ [Pan *et al.*, 2018]. Se ha reportado evidencia estructural sobre $eeNav1.4\alpha/eeNav\beta 1$ [Yan *et al.*, 2017], $hNav1.4\alpha/hNav\beta 1$ [Pan *et al.*, 2018] y $hNav1.7\alpha/hNav\beta 1$ [Shen *et al.*, 2019] acerca de un sitio de unión en la región TM de $Nav\beta 1$ entre las hélices S1 y S2 en el VSD DIII [Namadurai *et al.*, 2014; Zhu *et al.*, 2017]. Es de destacar que el hallazgo de las regiones TM de las

subunidades Nav β 1 y Nav β 3 indica que poseen una alta similitud de secuencia. Además, estudios *in vitro* revelan que la región TM de Nav β 3 se une de forma no covalente al VSD DIV [Zhu *et al.*, 2017] de manera similar como lo hace Nav β 1 [Yan *et al.*, 2017; Pan *et al.*, 2018; Shen *et al.*, 2019]. Sin embargo, no hay datos estructurales para señalar la ubicación del sitio de unión de Nav β 3 en las subunidades α .

Con frecuencia, se ha asumido que la subunidad Nav β 3 interactúa con la subunidad Nav α a través del mismo mecanismo que la subunidad Nav β 1 [Namadurai *et al.*, 2014]. No obstante, los estudios *in vitro* demuestran que tanto Nav β 1 como Nav β 3 atenúan la unión de la lidocaína a Nav1.3 α [Lenkowski *et al.*, 2003]. Dicho anestésico local se une a la hélice S6 del dominio IV mediante enlaces no covalentes [Ragsdale *et al.*, 1994] y evidencia estructural afirma que Nav β 1 forma PPI con Nav1.4 α en el IF-ECL S6 DIV [Yan *et al.*, 2017; Pan *et al.*, 2018; Shen *et al.*, 2019].

En esta investigación, se tienen bases teóricas, métodos y análisis de fuerte soporte, en donde se propone que el sitio de unión de Nav β 3 en la subunidad α formadora del poro es el mismo que el de Nav β 1. La suposición se basó en los análisis de datos estructurales de eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017] alineados en 3D con hNav1.4 α /hNav β 1 [Pan *et al.*, 2018]. El mismo patrón de conservación se encuentra nuevamente en el complejo (determinado en esta investigación) hNav1.4 α /hNav β 3 [Namadurai *et al.*, 2014; Pan *et al.*, 2018]. Además, se realizó el análisis y determinación de las PPI, propiedades de las ECLs e interfaz en todas las isoformas en las tres especies bajo un fuerte escrutinio (Tabla S1 y Figuras S1 a S9).

En las estructuras molde (3D) analizadas y nuestros modelos 3D, encontramos patrones de interacción en las interfaces Nav α /Nav β 1 y Nav α /Nav β 3. Tras la inspección minuciosa, es seguro generalizar nuestros hallazgos detallados de que estos patrones de interacción divergen significativamente entre isoformas que cruzan la barrera de las especies. En particular, observamos patrones de PPI específicos entre Nav α con subunidades Nav β 1 o Nav β 3.

Aplicando las capacidades de alineación bidimensional (2D) con 3D en combinación para la superposición racional de proteínas, las estructuras molde de hNav β 1 [Pan *et al.*, 2018] y hNav β 3 [Namadurai *et al.*, 2014; Pan *et al.*, 2018], se alinearon como referencias espaciales (Figura 7b). A continuación, identificamos todos los residuos que interactúan

entre las estructuras molde hNav1.4 α /hNav β 1 [Pan *et al.*, 2018] y el modelo 3D hNav1.4 α /hNav β 3 [Namadurai *et al.*, 2014; Pan *et al.*, 2018] (Figura 6). La identificación fue asistida por los residuos identificados inicialmente en la estructura cryo-EM eeNav1.4 α /eeNav β 1 [Yan *et al.*, 2017] como una referencia de alto valor de información para identificar la conservación en organismos relacionados de manera cercana o lejana, aquí: tres especies de mamífero *versus* anguila (Figura 5). Como un activo para la validación de las PPI, basadas no solo en las similitudes secuenciales sino también estructurales de hNav β 3 con hNav β 1 se encuentran significativamente por encima de la zona *twilight* (cf. Modelado molecular *in silico*) con $\approx 50\%$ y RMSD ≈ 1.2 , respectivamente. Tomados en conjunto todos los patrones topológicos, la quimiometría reveló pruebas convincentes (teóricas), las cuales indican que la subunidad Nav β 3 se une y modula Nav α en la misma posición y a través del mismo mecanismo que la subunidad Nav β 1, debido a que el patrón de residuos que interactúan es casi el mismo (cf. modelo 3D de hNav1.4 α /Nav β 3 y PDB 6AGF [Pan *et al.*, 2018] con hNav1.4 α /Nav β 1). Sin embargo, existen variaciones en los residuos conservados de PPI en ambas proteínas Nav α /Nav β (Figuras S1 – S10), los cuales podrían explicar hasta cierto punto las diferencias en la cinética de los Navs que deben confirmarse en estudios futuros *in silico* de dinámicas moleculares con investigaciones fundamentales experimentales *in vitro*.

Como inferencia del análisis de datos del estudio quimiométrico realizado, también estudiamos las estructuras conocidas de hNav β 2 [Das *et al.*, 2016] y hNav β 4 [Gilchrist *et al.*, 2013], en donde nuevamente se llevaron a cabo alineaciones 2D y 3D *versus* la estructura de referencia hNav β 1 (cf. Tabla 4 y Figuras 7c y 7d). Los datos críticos analizados de los alinamientos 2D y 3D de hNav β 2 *vs* hNav β 1 y hNav β 4 *vs* hNav β 1 se encuentran claramente dentro de la zona de *twilight* con $\approx 18,8$ y $18,1\%$, RMSD: 4,52 y 7,12, respectivamente. El grado de desajustes posicionales de residuos equivalentes se vuelve obvio a la vista, con respecto a las secuencias β 2 o β 4 frente a los residuos interactivos conocidos de la referencia hNav β 1 [Pan *et al.*, 2018]. La pérdida de posiciones conservadas PPI para los residuos de interacción de ambas subunidades (β 2, β 4) sugiere fuertemente la existencia de una interfaz totalmente distinta en un sentido estricto de interacción respecto a las ECLs y dominios con la subunidad α formadora de poro. Este hallazgo no ha sido reportado en la literatura, a pesar de la evidencia cryo-EM hNav β 2/hNav1.2 α [Pan *et al.*, 2019].

En esta etapa, se logró caracterizar el comportamiento topológico de las nueve isoformas y agruparlas en vista de sus distintos patrones de interacción (cf. Tablas 3 y S1, Figuras S1 – S10): (i) las isoformas hNav, mNav y rNav (1.2, 1.4 y 1.6), presentan ocho PPI con Nav β 1 y Nav β 3; (ii) las isoformas hNav, mNav y rNav (1.1, 1.3 y 1.7) presentan siete PPI. Ambos comparten características esenciales, por lo que sugerimos que tengan un efecto similar sobre la modulación. En contraste (débil), (iii) las isoformas hNav, mNav y rNav (1.1, 1.3, 1.5, 1.7, 1.8 y 1.9) coinciden en un residuo que no interactúa en la posición 6 en ECL S6 DIV. En contraste (fuerte), (iv) las isoformas hNav, mNav y rNav (1.5, 1.8 y 1.9) siempre comparten dos características: primero, todas carecen de dos o más PPI y, en segundo lugar, todas pertenecen a la clasificación farmacológica TTX-R, y se asocian a una carencia de interacción en la IF-ECL S1-S2 DIII, (v) mientras que las isoformas hNav, mNav y rNav (1.9) no interactúan en absoluto con la IF-ECL S6 DIV. Resumiendo, los hallazgos en una idea mecanicista, la activación de las isoformas que pertenecen a la clasificación farmacológica TTX-R, también podría coincidir en un mecanismo de modulación común. (vi) La ausencia de interacción en el PPI-Id 6 en la ECL S6 DIV concierne a las isoformas: (h, m, r) Navs (1.1, 1.3, 1.5, 1.7, 1.8 y 1.9). Este sitio de interacción (PPI-Id 6) expone una fuerte interacción por puente de sal para las isoformas h, m, rNav 1.2, 1.4 y 1.6. (Figuras S1 a S9). Se infiere que la presencia o ausencia de esa fuerte señal electrostática en la IF podría ser una característica significativa inherente para desencadenar variaciones dependientes de las isoformas en la modulación de poros y posiblemente una característica fundamental para la expresión en ciertos tejidos, la cual en la actualidad aun no es clara.

9.2. Aceleración hipotética de la inactivación rápida del gating de los Navs.

La despolarización de la membrana celular está asociada con un movimiento en espiral hacia la región extracelular para exponer los sensores de voltaje helicoidales S4 en la subunidad formadora del poro (Figuras 3 y 8). La exposición de los S4 a la superficie implica un cambio conformacional del canal para entrar en un estado abierto [Payandeh *et al.*, 2011; Long *et al.*, 2005 y 2007; Yu *et al.*, 2005]. Como resultado, la puerta de inactivación de tres aminoácidos (IFM) ubicada entre las hélices S6 DIII y S1 DIV se conecta rápidamente al poro y evita la entrada de iones de sodio (ocurre el bloqueo hidrófobo del poro desde la región intracelular y una congestión en la cavidad del poro por afluencia de Na⁺), lo que

conduce a un estado inactivado. Gracias a la identificación de residuos que interactúan en base a los modelos topológicos (secuenciales) y estructurales (espaciales) sistemáticos en esta investigación, fue posible vincularlos con los aspectos electrofisiológicos informados sobre la transición funcional de la activación a la inactivación de las nueve isoformas de Nav (Figuras 3, 8 y 9) [Catterall, 2012; Vargas *et al.*, 2012. En este estudio, es imprescindible mencionar que la glicosilación variable ligada a los residuos N de los ectodominios (IgD) no afecta a nuestros resultados quimioinformáticos ya que no se encuentra en la interfaz entre ambas subunidades [Namadurai *et al.*, 2015]. Además, Ferrera *et al.* [2006], demostraron que Nav β 1 puede cambiar las cargas electrostáticas en la superficie formadora del poro que interviene en la activación de los canales. También, Namadurai *et al.*, 2015 asumen que si una subunidad β se acerca lo suficiente a un sensor de voltaje podría ocurrir un cambio hiperpolarizante, i. e. en la membrana celular la mayoría de los Navs se inactivarían (rápidamente) aboliendo la corriente de Na⁺. Finalmente, Zhu *et al.*, [2017], concluyeron de sus estudios *in vitro* que Nav β 1 y Nav β 3 aceleran la inactivación al tener contacto con el S4 en Nav1.5 α , específicamente: Nav β 1 en S4 (DIII y DIV) y Nav β 3 en S4 DIII.

Curiosamente, en los modelos construidos y simulados Nava/Nav β 1 y Nava/Nav β 3 (Tabla S1 y Figuras S1 a S10), ambas subunidades β están ubicadas en las proximidades del sensor de voltaje S4 en DIII. En términos hipotéticos, cuando la membrana celular se despolariza, el S4 DIII se desplaza hacia la zona extracelular (hacia la zona extracelular y consecuentemente hacia las Nav β 1 ó Nav β 3), posteriormente, dos residuos conservados cargados positivamente en el S4 DIII entran en contacto repulsivo con una lisina (carga-carga electropositiva) conservada en ambas subunidades β (Tabla 5). Las tres cargas positivas están claramente expuestas en la superficie accesible al solvente de la superficie de la proteína formadora del poro (Figura 8). Es seguro concluir que se activa una fuerte señal bioquímica cuando esta repulsión de carga tiene lugar en el S4 DIII (Figura 8). Como consecuencia directa de la geometría general del canal, se produce un cambio en el dominio de las IgD (se produce un movimiento hasta la fecha desconocido de los ectodominios de Nav β 1 y Nav β 3) para iniciar la modulación (el mecanismo hipotético de la modulación de la inactivación rápida por Nav β 1 ó Nav β 3). El segmento S1-S2 DIII sirve como una bisagra flexible para formar una asociación no covalente con las IgD β 1 y β 3. En una respuesta más lenta a la despolarización los S5 DI y S5 DIV terminan el ciclo forzando a los sensores de

voltaje S4 (DI y DIV) a regresar a su posición inicial en forma de resorte (generando un cambio conformacional en el canal hasta la fecha desconocido). Finalmente, el S6 DIV detiene —como un ancla— ambas subunidades β . Sobre una base teórica, los presentes hallazgos explican la cooperación entre las ECLs y ciertas subunidades Nav β para la modulación del canal. Desde un punto de vista evolutivo, esto tiene sentido, ya que las llamadas proteínas externas (β s auxiliares) para ayudar a modular la función de la proteína (poro α) se logra mucho más rápido que la adaptación de segmentos de asas proteicas mediante la selección evolutiva lenta de mutaciones puntuales aleatorias sobre las asas. Evidentemente, esto sucedió en algún punto en el tiempo de la evolución celular en el que ya se había producido la fusión de genes para transformar el canal homotetramérico (bacteriano) monocatenario en una proteína de poro, que se compone de cuatro dominios diferentes (“subunidad monocatenaria hetero-tetra-dominio”). En este contexto, la existencia de nueve mutantes estrechamente relacionadas (isoformas) muestran el trabajo de la evolución desde el pasado “cercano” hasta el presente, cuando una proteína ancestral común ha ido evolucionando junto con la expresión génica diferencial y la especialización tisular.

9.3. Propiedades de los residuos de las ECLs en las isoformas Navas

El S5 DI constituye la ECL más larga. Su tamaño aberrante refleja mayor interfaz en Nav β 1 y Nav β 3, el cual presenta un segundo PPI; PPI-Id 1 que se conserva en todas las isoformas y especies; PPI-Id 2 se conserva en casi todas las isoformas con la excepción de Nav1.8 α y Nav1.9 α . La longitud del ECL de S5 DIV contiene un residuo cargado negativamente (PPI-Id 5) que se conserva en las nueve isoformas de las tres especies. Posiblemente, las ECLs de S5 DI y DIV han evolucionado de manera constante en contacto con las subunidades Nav β 1 y Nav β 3, lo que lleva a especulaciones sobre su papel como principales sitios de unión para la modulación de Nav α .

Las propiedades quimiométricas que describen a los S5 y S6 se documentan en las Figuras 11, 12 y 16. En las tres especies, Nav1.4 α (respecto a la isoforma Nav1.5 α) representa la cantidad más alta de residuos polares y cargados negativamente en S5 y S6. Dos isoformas contienen el menor número de residuos apolares: Nav1.8 α y Nav1.9 α . Ambos muestran el mayor porcentaje de residuos de ECR polares y, en particular, en ECL S5 y S6 para las tres especies. Dicho hallazgo refleja la literatura existente que especula sobre la

ausencia de evidencia experimental para la cooperación de Nav1.9 α con las subunidades β para la modulación del *gating*. Las Figuras 11, 12 y 16 resume las propiedades de las ECLs. Nav α 1.4 y Nav α 1.6 poseen el mismo número de PPI. A pesar de este patrón común, las diferencias sutiles también pueden explicar por qué difiere su comportamiento cinético. El ECL de S5 en DI en Nav1.4 α (respecto a Nav α 1.6) aloja la cantidad más alta (o más pequeña) de aminoácidos polares y cargados negativamente. Curiosamente, la isoforma Nav1.6 α acumula residuos aún más polares y cargados negativamente en sus asas S5 en DIII. Los dominios DI y DIII se enfrentan entre sí desde posiciones diametralmente opuestas a través de la parte del poro central. Ambos han desarrollado las ECL S5 más largas entre todas las Nav α . Se puede especular que sus longitudes podrían reflejar la principal atracción electrostática de Nav α para la conducción de Na⁺ hacia el poro del canal. Este equilibrio de residuos distribuidos en las ECLs S5 (DI y DIII) probablemente ha evolucionado para proporcionar a las isoformas Nav1.4 α y Nav1.6 α una MEPS similar a una huella digital con características inherentes para realizar una función específica en el tejido. Hasta ahora, los hallazgos mencionados anteriormente no se han informado en la literatura existente.

9.4. Propiedades de volumen y superficie de las ECLs en las isoformas Navas

La evolución conduce a mutaciones puntuales aleatorias o SNPs con consecuencias variables en la supervivencia de los organismos, en donde a nivel molecular cambian las estructuras y funciones de las proteínas [Kinch y Grishin, 2002; Starr y Thornton, 2016]. Las isoformas pueden entenderse como estados transitorios durante la evolución divergente para separarlos de una proteína ancestral común cuando las células evolucionan a tejidos más especializados en los organismos [Kauvar y Villar, 1998; Russell y Sasieni, 1998]. En proteínas de membrana, la señalización bioquímica no se encuentra rara vez en los segmentos de las asas expuestas en las superficies celulares con una conservación notable de residuos relevantes para la señal en medio de los segmentos de asas variables. Esta observación es la razón fundamental para combinar técnicas de alineación 2D y 3D que nos permiten revelar este micromundo oculto de aminoácidos de señalización o interacción que subyacen en la interfaz α/β [Spitzer *et al.*, 2013; Richards, 1977; Gainza *et al.*, 2019]. El tipo de estructura de la proteína, también conocida como unidad de pliegue, está más conservada que su secuencia primaria subyacente. Además, las estructuras inalteradas mantienen la

función bioquímica, lo que a veces se puede observar incluso en casos extremos de divergencia de secuencia [Illergård *et al.*, 2009; Sousounis *et al.*, 2012; Gibrat *et al.*, 1996 95-97]. Es de destacar que cada ECR tiene 16 ECL en cada isoforma.

Con respecto a las nueve isoformas, la isoforma Nav1.4 α tiene el mayor volumen y la SAA más amplia respecto a las ECR de las isoformas en general. Además, respecto a la arquitectura del poro, este patrón se repite para las ECLs S5 y S6. En contraste, la isoforma Nav1.9 α tiene el volumen y SAA más pequeños en la ECR y las ECLs S5 y S6 (Figuras 14 y 15). Este hallazgo aún no ha sido reportado por otros autores.

La IF-ECL S5 DI en h, m, rNav1.4 α tiene un volumen molecular mucho mayor que las otras isoformas. Por el contrario, el volumen molecular de las ECLs S6 DIV en Nav1.9 α es significativamente menor que en todas las demás isoformas. Esta propiedad quimiométrica explica muy bien por qué las isoformas de Nav1.9 α no entran en contacto con ambas subunidades β (1 y 3) a través de la ECL S6 DIV, cabe mencionar que tampoco ha sido reportado por otros autores.

De todas las isoformas, las isoformas h, m, rNav1.4 α (h, m, rNav1.8 α y h, m, rNav1.9 α) tienen las más altas N-MEPS en las IF-ECLs (Figura 16). Estos patrones son idénticos para las ECLs S5 DI y S6 DIV. Por otro lado, Nav1.8 α y Nav1.9 α tienen los P-MEPS más altos para las ECL de S1-S2 DIII. Curiosamente, las isoformas Nav1.4 α poseen las áreas más pequeñas de N-MEPS en las ECLs S1-S2 DIII (Figura S17). Los rasgos característicos dependientes de la isoforma de cada isoforma podrían atribuirse a la afinidad de la atracción electrostática a las subunidades Nav β 1 y Nav β 3 en donde predomina en la interfaz la electronegatividad MEPS.

9.5. Propiedades de interfaz de Nava/Nav β 1 y Nava/Nav β 3

La observación de que sólo una pequeña parte de la superficie total pertenece a segmentos estructurales-funcionales se refleja en la alta conservación de dichos segmentos [Caffrey *et al.*, 2004; Grishin y Phillips, 1994; Binkowski y Joachimiak, 2008; Marshall *et al.*, 2002]. Especialmente, las fuerzas electrostáticas a menudo actúan como determinantes críticos para la señalización bioquímica u otras funciones proteicas como el reconocimiento de ligandos, las afinidades o la estabilidad estructural. Se menciona a menudo, que las PPIs tienen lugar en ubicaciones específicas de la superficie (huellas digitales) con

complementariedad geométrica y química [Chothia y Janin, 1975; Sheinerman *et al.*, 2000; Heifetz *et al.*, 2002; Vizcarra y Mayo, 2005]. De esta manera, las ECLs en el canal de sodio tienden a mantener similitudes fisicoquímicas en sus superficies, lo cual se resume a distintos patrones de interacción.

El área de superficie variable entre las IF-ECL y ambas subunidades β se documentó en la Tabla S2 y Figura 17. Para la mayoría de las isoformas de los Navs, el siguiente patrón de interacción de las IF-ECLs con las Navs en la interfaz general de área enterrada se mantiene en el siguiente orden: S5 DI>S1-S2 DIII>S6 DIV>S5 DIV, a excepción de el área de interfaz de la IF-ECL S6 DIV/h,m,rNav1.9, en donde los resultados indican que las áreas de interfaz enterrada y la superficie de los átomos que forman el área de interfaz enterrada son mucho más pequeñas en comparación con todas las demás isoformas (como consecuencia del tamaño reducido del volumen y superficie molecular de la IF-ECL S6 DIV, cf. Apartado 9.4 y Figuras 14 y 15). Dicha observación está en excelente consonancia con el papel electrofisiológico de h, m, rNav1.9 α , que carece de PPI con ambas subunidades $\beta 1$ y $\beta 3$ (cf. Apartados 8.1, 9.3, 9.4, Tablas 3 y S1 y Figuras S1 a S10) Por lo tanto, no parece extraño inferir que las isoformas de mamífero Nav1.9 α no interactúan con la IF-ECL S6 DIV.

Las Figuras 18 a 21 informan en detalle sobre las puntuaciones porcentuales relativas de las IF-ECLs y ambas subunidades β para las siguientes propiedades: PSA, NPSA, P-MEPS y N-MEPS, medidas en una escala de superficie atómica, que se restringe a la superficie accesible al solvente de los átomos que forman la interfaz (no área de interfaz enterrada). Las isoformas Nav1.2 α y Nav1.6 α poseen patrones de interacción, que se asemejan a los de las IF-ECL S5 DI, S1-S2 DIII y S5 DIV. Estas características (no reportadas hasta la fecha) podrían explicar por qué las propiedades de la superficie de la interfaz en ambas subunidades (Nav α y Nav β) se conservan ya que reflejan una modulación similar (de acuerdo a los PPI-Ids del *gating* hipotético cf. Tablas 3 y S1, Figuras 9 y S1 a S9). Curiosamente, Nav1.2 α y Nav1.6 α presentan los mismos sitios PPI junto con Nav1.4 α . Sin embargo, el tamaño, las propiedades residuos, SAA, MEPS, el volumen molecular, difieren bastante entre Nav1.2 α o Nav1.6 α *versus* Nav1.4 α , enfatizando las IF-ECLs S5 DI. Por otro lado, las IF-ECL S6 DIV no muestran ninguna similitud entre las isoformas, lo cual está en línea con el papel fundamental de las IF-ECL S6 DIV como un fuerte contribuyente a la modulación de poros debido a sus fuerzas electrostáticas únicas en su superficie.

10. CONCLUSIONES

En este trabajo, se analizaron las interfaces proteína-proteína observadas en estructuras determinadas por cryo-EM y postulamos otras en modelos derivados de estructuras molde 3D determinadas experimentalmente (cryo-EM y estructuras cristalográficas). Los modelos se generaron para las nueve isoformas existentes de tres especies (humano, rata, ratón). La interfaz es referente a los residuos de asas extracelulares en estrecho contacto con las subunidades Nav β 1 o Nav β 3. Gracias al análisis quimiométrico, se formuló un modelo para determinar la aceleración de la inactivación rápida del *gating* en las subunidades formadora de poros Nav α modulada por la presencia de proteínas auxiliares Nav β 1 o Nav β 3. En el terreno teórico, obtuvimos una visión mecanicista de los movimientos alrededor del sensor de voltaje S4 DIII, que es modulado (teóricamente) por las subunidades β .

Describimos la modulación de las Navas en términos de un modelo esquemático de PPI entre la proteína α transmembrana formadora del poro y las proteínas β auxiliares para las nueve isoformas en tres especies de mamíferos. Nuestros modelos estructurales y el análisis topológico de secuencias llevan a la conclusión de que sus distintas interfaces reflejan las diferencias observadas en la cinética de activación.

Calculamos patrones quimiométricos para criterios específicos, como enlaces no covalentes, longitud de asas, área accesible al solvente, superficies enterradas, volúmenes moleculares y otros descriptores electrostáticos. Las isoformas se agruparon de acuerdo a patrones de interacción comunes y se opusieron a otras con patrones diferentes, y todos los resultados se relacionaron mecánicamente con informes sobre la cinética de activación e inactivación rápida. Los patrones incluyen un área accesible al solvente o posiciones conservadas para residuos cargados de manera opuesta a ambos lados de la interfaz. Estos hallazgos sobre variaciones sutiles en los patrones electrostáticos afectan la capacidad de modulación individual de cada isoforma, lo cual está en consonancia con las observaciones electrofisiológicas de la cinética del *gating* ampliamente estudiadas y resumido gráficamente en nuestros dibujos topológicos esquemáticos, simulaciones *in silico*, análisis estructurales y una gran cantidad de datos descriptores de información quimiométrica en la interfaz.

Este estudio quimioinformático se basó completamente en observaciones tomadas de la literatura existente, y nuestros resultados están en línea con sus hallazgos experimentales. Además, informamos dos patrones de interacción en las PPI-ID no identificados hasta ahora

para las estructuras molde 3D (estructuras cryo-EM), así como los modelos de interfaz propuestos, los cuales encajan en una imagen mecanicista más amplia con los otros PPI-ID de interacción, que fueron reportados por primera vez por Yan *et al.*, con el advenimiento de una estructura completa de canales de sodio para las especies de vertebrados (anguila) [Yan *et al.*, 2017].

Este trabajo en su mayor parte heurístico con suposiciones y bases teóricas enfocado en análisis estructurales de residuos e interfaces, concluye con los siguientes puntos convergentes:

- Las isoformas Nav1.5 α , Nav1.8 α y Nav1.9 α presentan la menor cantidad de PPIs con las subunidades Nav β 1 y Nav β 3, en donde podrían influir en la mecánica funcional del *gating*, participando con menor grado en la inactivación rápida de la cinética de los Navs. Cabe destacar que son isoformas resistentes a TTX y presentan los valores más bajos de superficie accesible al solvente de interfaz (área enterrada y área de átomos que forman la interfaz) con Nav β 1 y Nav β 3, el cual puede ser un indicio de la afinidad de dicha toxina a los Navs.
- Las PPI-Ids que se forman en las interfaces Nav α /Nav β 1 y Nav α /Nav β 3 podrían ser un indicio de la mecánica funcional y estructural del *gating* de acuerdo al número de PPI en las isoformas.
- Los residuos que participan en las PPI de las subunidades Nav β 3 en las tres especies tienen posiciones estructurales casi idénticas a las subunidades Nav β 1, por lo que podrían modular a Nav α de manera muy similar.
- Los residuos que participan en las PPI-Ids de las asas S5 DI y S5 DIV (PPI-Ids 1 y 5, respectivamente), se encuentran conservados en todas las isoformas y especies. Además, están implicados en la modulación del *gating* hipotética de los Navs, los cuales ayudan a regresar más rápidamente a los sensores de voltaje S4 (DI y D), podría conducir a cerrar más rápidamente a la subunidad formadora del poro dando lugar a la aceleración de la inactivación rápida.
- El trímero determinado experimentalmente de Na ν β 3 [Namadurai *et al.*, 2014] alineado estructuralmente a Na ν β 1 en el complejo de la interfaz Na ν β 1/Nav α forma un empalme molecular. En cambio, un monómero de la subunidad Na ν β 3 se alinea

con alta similitud a $\text{Na}_v\beta 1$, en donde es plausible discernir que el sitio de interacción de la interface de $\text{Na}_v\beta 3$ con $\text{Nav}\alpha$ es posiblemente el mismo que ocupa $\text{Na}_v\beta 1$ en la interfaz $\text{Na}_v\beta 1/\text{Nav}\alpha$.

- La superficie de interfaz del asa extracelular S6 DIV de la isoforma $\text{Nav}1.9\alpha$ y las subunidades $\text{Nav}\beta 1$ y $\text{Nav}\beta 3$ en las tres especies, no presenta interfaz debido a la longitud, área, volumen molecular menores y características electrostáticas diferentes respecto a las demás isoformas.

Finalmente, este estudio podría orientar a la investigación científica futura en biología molecular o ayudar a diseñar estudios de mutagénesis sitio dirigida en la interfaz de subunidades de los canales de sodio dependientes de voltaje. En particular, los estudios de dinámica molecular en supercomputadoras podrían simular las trayectorias de activación a lo largo del tiempo y confirmar que los movimientos de las isoformas se pueden agrupar siguiendo los patrones quimioinformáticos propuestos.

11. REFERENCIAS

- Agnew, W. S.; Levinson, S. R.; Brabson, J. S.; Raftery, M. A. Purification of the tetrodotoxin-binding component associated with the voltage-sensitive sodium channel from *Electrophorus electricus* electroplax membrane. *Proc. Natl. Acad. Sci.* **1978**, *75*, 2606–2610.
- Ahern, C.A.; Payandeh, J.; Bosmans, F.; Chanda, B. The hitchhiker's guide to the voltage-gated sodium channel galaxy. *J. Gen. Physiol.* **2015**, *147*, 1–24, doi:10.1085/jgp.201511492.
- Bagal, S.K.; E Marron, B.; Owen, R.M.; Storer, R.I.; A Swain, N. Voltage gated sodium channels as drug discovery targets. *Channels* **2015**, *9*, 360–366, doi:10.1080/19336950.2015.1079674.
- Baker, N.; Sept, D.; Joseph, S.; Holst, M.J.; McCammon, J.A. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Natl. Acad. Sci.* **2001**, *98*, 10037–10041, doi:10.1073/pnas.181342398.
- Binkowski, T.A.; Joachimiak, A. Protein Functional Surfaces: Global Shape Matching and Local Spatial Alignments of Ligand Binding Sites. *BMC Struct. Boil.* **2008**, *8*, 45, doi:10.1186/1472-6807-8-45.
- Bosmans, F.; Martin-Eauclaire, M.-F.; Swartz, K.J. Deconstructing voltage sensor function and pharmacology in sodium channels. *Nat.* **2008**, *456*, 202–208, doi:10.1038/nature07473.
- Brackenbury, W.J.; Isom, L.L. Na⁺ Channel? Subunits: Overachievers of the Ion Channel Family. *Front. Pharmacol.* **2011**, *2*, 53, doi:10.3389/fphar.2011.00053.
- Buffington, S.A.; Rasband, M.N. Na⁺ channel-dependent recruitment of Nav β 4 to axon initial segments and nodes of Ranvier. *J. Neurosci.* **2013**, *33*, 6191–202, doi:10.1523/JNEUROSCI.4051-12.2013.
- Caffrey, D.R.; Somaroo, S.; Hughes, J.D.; Mintseris, J.; Huang, E.S. Are protein–protein interfaces more conserved in sequence than the rest of the protein surface? *Protein Sci.* **2004**, *13*, 190–202, doi:10.1110/ps.03323604.
- Calhoun J.D.; Isom L.L. The Role of Non-Pore-Forming β Subunits in Physiology and Pathophysiology of Voltage-Gated Sodium Channels. *Handb Exp Pharmacol.* **2014**, *221*: 51-89, doi:10.1007/978-3-642-41588-3_4.
- Capes, D.L.; Goldschen-Ohm, M.P.; Arcisio-Miranda, M.; Bezanilla, F.; Chanda, B. Domain IV voltage-sensor movement is both sufficient and rate limiting for fast inactivation in sodium channels. *J. Gen. Physiol.* **2013**, *142*, 101–112, doi:10.1085/jgp.201310998.
- Catterall, W.A. From Ionic Currents to Molecular Mechanisms. *Neuron* **2000**, *26*, 13–25, doi:10.1016/s0896-6273(00)81133-2.
- Catterall, W.A. Neurotoxins that act on voltage-sensitive sodium channels in excitable membranes. *Annu Rev Pharmacol Toxicol.* **1980**, *20*, 15–43, doi: 10.1126/stke.2532004re15
- Catterall, W.A. Sodium Channels, Inherited Epilepsy, and Antiepileptic Drugs. *Annu. Rev. Pharmacol. Toxicol.* **2014**, *54*, 317–338, doi:10.1146/annurev-pharmtox-011112-140232.
- Catterall, W.A. Structure and function of voltage-gated ion channels. *Annu. Rev. Biochem.* **1995**, *64*, 493–531.
- Catterall, W.A. Voltage-gated sodium channels at 60: structure, function and pathophysiology. *J. Physiol.* **2012**, *590*, 2577–2589, doi:10.1113/jphysiol.2011.224204.
- Catterall, W.A.; Cestele, S.; Yarov-Yarovoy, V.; Yu, F.H.; Konoki, K.; Scheuer, T. Voltage-gated ion channels and gating modifier toxins. *Toxicon* **2007**, *49*, 124–141, doi:10.1016/j.toxicon.2006.09.022.
- Catterall, W.A.; Goldin, A.L.; Waxman, S.G. International Union of Pharmacology. XLVII. Nomenclature and Structure-Function Relationships of Voltage-Gated Sodium Channels. *Pharmacol. Rev.* **2005**, *57*, 397–409, doi:10.1124/pr.57.4.4.
- Cestèle, S. and Catterall, W.A. Molecular mechanisms of neurotoxin action on voltage-gated sodium channels. **2000**. *Biochimie*, *82*, 883–892.
- Chanda, B.; Bezanilla, F. Tracking Voltage-dependent Conformational Changes in Skeletal Muscle Sodium Channel during Activation. *J. Gen. Physiol.* **2002**, *120*, 629–645, doi:10.1085/jgp.20028679.
- Chen, C.; Calhoun, J.D.; Zhang, Y.; Lopez-Santiago, L.; Zhou, N.; Davis, T.H.; Salzer, J.L.; Isom, L.L. Identification of the Cysteine Residue Responsible for Disulfide Linkage of Na⁺Channel α and β 2 Subunits. *J. Biol. Chem.* **2012**, *287*, 39061–39069, doi:10.1074/jbc.m112.397646.

- Chen, C.; Cannon, S.C. Modulation of Na⁺ channel inactivation by the β 1 subunit: A deletion analysis. *Pflügers Archiv - European Journal of Physiology* **1995**, *431*, 186–195, doi:10.1007/bf00410190.
- Chopra, S.S.; Watanabe, H.; Zhong, T.P.; Roden, D.M. Molecular cloning and analysis of zebrafish voltage-gated sodium channel beta subunit genes: implications for the evolution of electrical signaling in vertebrates. *BMC Evol. Boil.* **2007**, *7*, 113, doi:10.1186/1471-2148-7-113.
- Chothia, C.; Janin, J. Principles of protein–protein recognition. *Nat.* **1975**, *256*, 705–708, doi:10.1038/256705a0.
- Claes, L.R.; Deprez, L.; Suls, A.; Baets, J.; Smets, K.; Van Dyck, T.; Deconinck, T.; Jordano, A.; De Jonghe, P. The SCN1A variant database: A novel research and diagnostic tool. *Hum. Mutat.* **2009**, *30*, E904–E920.
- Clustal Omega. Available online. *Met. Powder Rep.* **2008**, *63*, 11, doi:10.1016/s0026-0657(08)70101-2.
- Cukuroglu, E.; Gursoy, A.; Keskin, O. HotRegion: a database of predicted hot spot clusters. *Nucleic Acids Res.* **2011**, *40*, D829–D833, doi:10.1093/nar/gkr929.
- Cusdin, F.S.; Clare, J.J.; Jackson, A.P. Trafficking and Cellular Distribution of Voltage-Gated Sodium Channels. *Traffic* **2008**, *9*, 17–26, doi:10.1111/j.1600-0854.2007.00673.x.
- Das, S.; Gilchrist, J.; Bosmans, F.; Van Petegem, F. Binary architecture of the Nav1.2- β 2 signaling complex. *Elife* **2016**, *5*, e10960.
- Dolinsky, T.J.; Czodrowski, P.; Li, H.; Nielsen, J.E.; Jensen, J.H.; Klebe, G.; Baker, N. PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations. *Nucleic Acids Res.* **2007**, *35*, W522–W525, doi:10.1093/nar/gkm276.
- E Patton, D.; Isom, L.L.; Catterall, W.; Goldin, A.L. The adult rat brain beta 1 subunit modifies activation and inactivation gating of multiple sodium channel alpha subunits. *J. Boil. Chem.* **1994**, *269*, 17649–17655.
- Ezkurdia, I.; Bartoli, L.; Fariselli, P.; Casadio, R.; Valencia, A.; Tress, M.L. Progress and challenges in predicting protein-protein interaction sites. *Briefings Bioinform.* **2008**, *10*, 233–246, doi:10.1093/bib/bbp021.
- Ferrera, L.; Moran, O. β 1 subunit modulates the Nav1.4 sodium channel by changing the surface charge. *Exp. Brain Res.* **2006**, *172*, 139–150, doi:10.1007/s00221-005-0323-4.
- Fux, J.E.; Mehta, A.; Moffat, J.; Spafford, J.D. Eukaryotic Voltage-Gated Sodium Channels: On Their Origins, Asymmetries, Losses, Diversification and Adaptations. *Front. Physiol.* **2018**, *9*, 9, doi:10.3389/fphys.2018.01406.
- Gainza, P.; Sverrisson, F.; Monti, F.; Rodolà, E.; Boscaini, D.; Bronstein, M.M.; Correia, B.E. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat. Methods* **2019**, *17*, 1–9, doi:10.1038/s41592-019-0666-6.
- Gibrat, J.-F.; Madej, T.; Bryant, S.H. Surprising similarities in structure comparison. *Curr. Opin. Struct. Boil.* **1996**, *6*, 377–385, doi:10.1016/s0959-440x(96)80058-3.
- Gilchrist, J.; Olivera, B.M.; Bosmans, F. Animal Toxins Influence Voltage-Gated Sodium Channel Function. *Voltage Gated Sodium Channels. Handbook of Experimental Pharmacology*, vol 221. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41588-3_10
- Gilchrist, J.M.; Das, S.; Van Petegem, F.; Bosmans, F. Crystallographic insights into sodium-channel modulation by the β 4 subunit. *Proc. Natl. Acad. Sci.* **2013**, *110*, E5016–E5024, doi:10.1073/pnas.1314557110.
- Glass, W.G.; Duncan, A.L.; Biggin, P.C. Computational Investigation of Voltage-Gated Sodium Channel β 3 Subunit Dynamics. *Front. Mol. Biosci.* **2020**, *7*, 40, doi:10.3389/fmolb.2020.00040.
- Goddard, T.D.; Huang, C.C.; Meng, E.C.; Pettersen, E.F.; Couch, G.S.; Morris, J.H.; Ferrin, T.E. UCSF ChimeraX: Meeting modern challenges in visualization and analysis. *Protein Sci.* **2017**, *27*, 14–25, doi:10.1002/pro.3235.
- Goldin, A. L. Diversity of mammalian voltage-gated sodium channels. *Ann. N. Y. Acad. Sci.* **1999**, *868*, 38–50.

- Goldin, A.L.; Barchi, R.L.; Caldwell, J.H.; Hofmann, F.; Howe, J.R.; Hunter, J.C.; Kallen, R.G.; Mandel, G.; Meisler, M.H.; Netter, Y.B. Nomenclature of voltage-gated sodium channels. *Neuron* **2000**, *28*, 365–368, doi:10.1016/s0896-6273(00)00116-1.
- Grishin, N.V.; Phillips, M.A. The subunit interfaces of oligomeric enzymes are conserved to a similar extent to the overall protein sequences. *Protein Sci.* **1994**, *3*, 2455–2458, doi:10.1002/pro.5560031231.
- Heifetz, A.; Katchalski-Katzir, E.; Eisenstein, M. Electrostatics in protein–protein docking. *Protein Sci.* **2002**, *11*, 571–587.
- Hermann, J.C.; Marti-Arbona, R.; Fedorov, A.A.; Fedorov, E.; Almo, S.C.; Shoichet, B.K.; Raushel, F.M. Structure-based activity prediction for an enzyme of unknown function. *Nat.* **2007**, *448*, 775–779, doi:10.1038/nature05981.
- Hodgkin, A.L.; Huxley, A.F. Propagation of electrical signals along giant nerve fibers. *Proc R Soc Lond B Biol Sci.* **1952**, *140*, 177–83.
- Huang, W.; Liu, M.; Yan, S.; Yan, N. Structure-based assessment of disease-related mutations in human voltage-gated sodium channels. *Protein Cell* **2017**, *8*, 401–438, doi:10.1007/s13238-017-0372-z.
- Illergård, K.; Ardell, D.H.; Elofsson, A. Structure is three to ten times more conserved than sequence—A study of structural response in protein cores. *Proteins: Struct. Funct. Bioinform.* **2009**, *77*, 499–508, doi:10.1002/prot.22458.
- Islas, A.A.; Sánchez-Solano, A.; Scior, T.; Millan-Perezpeña, L.; Salinas-Stefanon, E.M. Identification of Navβ1 Residues Involved in the Modulation of the Sodium Channel Nav1.4. *PLOS ONE* **2013**, *8*, e81995, doi:10.1371/journal.pone.0081995.
- Isom, L.L. Sodium channel beta subunits: anything but auxiliary. *Neuroscientist* **2001**, *7*, 42–54.
- Isom, L.L.; De Jongh, K.S.; Patton, D.E.; Reber, B.F.; Offord, J.; Charbonneau, H.; Walsh, K.; Goldin, A.L.; Catterall, W.A. Primary structure and functional expression of the β1 subunit of the rat brain sodium channel. *Science* **1992**, *256*, 839–842.
- Isom, L.L.; Ragsdale, D.S.; De Jongh, K.S.; E Westenbroek, R.; Reber, B.F.; Scheuer, T.; A Catterall, W. Structure and function of the beta 2 subunit of brain sodium channels, a transmembrane glycoprotein with a CAM motif. *Cell* **1995**, *83*, 90121–90123.
- Isom, L.L.; Scheuer, T.; Brownstein, A.B.; Ragsdale, D.S.; Murphy, B.J.; Catterall, W.A. Functional Co-expression of the 1 and Type IIA Subunits of Sodium Channels in a Mammalian Cell Line. *J. Boil. Chem.* **1995**, *270*, 3306–3312, doi:10.1074/jbc.270.7.3306.
- Izidoro, S.; De Melo-Minardi, R.C.; Pappa, G.L. GASS: identifying enzyme active sites with genetic algorithms. *Bioinform.* **2014**, *31*, 864–870, doi:10.1093/bioinformatics/btu746.
- Jacobson, M.P.; Kalyanaraman, C.; Zhao, S.; Tian, B. Leveraging structure for enzyme function prediction: methods, opportunities, and challenges. *Trends Biochem. Sci.* **2014**, *39*, 363–71, doi:10.1016/j.tibs.2014.05.006.
- Jiang, D.; Shi, H.; Tonggu, L.; El-Din, T.M.G.; Linaeus, M.J.; Zhao, Y.; Yoshioka, C.; Zheng, N.; Catterall, W.A. Structure of the Cardiac Sodium Channel. *Cell* **2020**, *180*, 122–134.e10, doi:10.1016/j.cell.2019.11.041.
- Kauvar, L.M.; O Villar, H. Deciphering cryptic similarities in protein binding sites. *Curr. Opin. Biotechnol.* **1998**, *9*, 390–394, doi:10.1016/s0958-1669(98)80013-x.
- Keskin, O.; Ma, B.; Nussinov, R. Hot Regions in Protein–Protein Interactions: The Organization and Contribution of Structurally Conserved Hot Spot Residues. *J. Mol. Boil.* **2005**, *345*, 1281–1294, doi:10.1016/j.jmb.2004.10.077.
- Kinch, L.N. ; Grishin, N.V. Evolution of protein structures and functions. *Curr. Opin. Struct. Biol.* **2002**, *12*, 400–408.
- Kruger, L.C.; Isom, L.L. Voltage-Gated Na⁺ Channels: Not Just for Conduction. *Cold Spring Harb. Perspect. Boil.* **2016**, *8*, a029264, doi:10.1101/cshperspect.a029264.
- Lenkowski, P.W.; Shah, B.S.; E Dinn, A.; Lee, K.; Patel, M.K. Lidocaine block of neonatal Nav1.3 is differentially modulated by co-expression of β1 and β3 subunits. *Eur. J. Pharmacol.* **2003**, *467*, 23–30, doi:10.1016/s0014-2999(03)01595-4.

- Li, X.; Keskin, O.; Ma, B.; Nussinov, R.; Liang, J. Protein–Protein Interactions: Hot Spots and Structurally Conserved Residues often Locate in Complemented Pockets that Pre-organized in the Unbound States: Implications for Docking. *J. Mol. Boil.* **2004**, *344*, 781–795, doi:10.1016/j.jmb.2004.09.051.
- Lomize, M.A.; Lomize, A.L.; Pogozheva, I.D.; Mosberg, H.I. OPM: Orientations of Proteins in Membranes database. *Bioinform.* **2006**, *22*, 623–625, doi:10.1093/bioinformatics/btk023.
- Long, S.B. Crystal Structure of a Mammalian Voltage-Dependent Shaker Family K⁺ Channel. *Sci.* **2005**, *309*, 897–903, doi:10.1126/science.1116269.
- Long, S.B.; Tao, X.; Campbell, E.B.; MacKinnon, R. Atomic structure of a voltage-dependent K⁺ channel in a lipid membrane-like environment. *Nat.* **2007**, *450*, 376–382, doi:10.1038/nature06265.
- Ma, B.; Elkayam, T.; Wolfson, H.; Nussinov, R. Protein-protein interactions: Structurally conserved residues distinguish between binding sites and exposed protein surfaces. *Proc. Natl. Acad. Sci.* **2003**, *100*, 5772–5777, doi:10.1073/pnas.1030237100.
- Makita, N.; Bennett, P.B.; George, A.L. Jr. Molecular determinants of β 1 subunit-induced gating modulation in voltage-dependent Na⁺ channels. *J. Neurosci.* **1996**, *22*, 7117–7127. doi:10.1523/JNEUROSCI.16-22-07117.1996.
- Mantegazza, M.; Catterall, W.A. *Voltage-Gated Na⁺ Channels*; Oxford University Press (OUP), 2012; Vol. 80, pp. 41–54;.
- Marshall, S.A.; Morgan, C.S.; Mayo, S.L. Electrostatics significantly affect the stability of designed homeodomain variants. *J. Mol. Boil.* **2002**, *316*, 189–199, doi:10.1006/jmbi.2001.5326.
- McCormick, K.A.; Isom, L.L.; Ragsdale, D.; Smith, D.; Scheuer, T.; Catterall, W.A. Molecular Determinants of Na⁺ Channel Function in the Extracellular Domain of the β 1 Subunit. *J. Boil. Chem.* **1998**, *273*, 3954–3962, doi:10.1074/jbc.273.7.3954.
- McCormick, K.A.; Srinivasan, J.; White, K.; Scheuer, T.; Catterall, W.A. The Extracellular Domain of the β 1 Subunit Is Both Necessary and Sufficient for β 1-like Modulation of Sodium Channel Gating. *J. Boil. Chem.* **1999**, *274*, 32638–32646, doi:10.1074/jbc.274.46.32638.
- Messner, D.J.; A Catterall, W. The sodium channel from rat brain. Separation and characterization of subunits. *J. Boil. Chem.* **1985**, *260*, 10597–10604.
- Microsoft Corporation. Microsoft Excel. Available online: <https://office.microsoft.com/excel> (accessed on 2016).
- Morgan, K.; Stevens, E.B.; Shah, B.; Cox, P.J.; Dixon, A.K.; Lee, K.; Pinnock, R.D.; Hughes, J.; Richardson, P.J.; Mizuguchi, K.; et al. beta 3: An additional auxiliary subunit of the voltage-sensitive sodium channel that modulates channel gating with distinct kinetics. *Proc. Natl. Acad. Sci.* **2000**, *97*, 2308–2313, doi:10.1073/pnas.030362197.
- Namadurai, S.; Balasuriya, D.; Rajappa, R.; Wiemhöfer, M.; Stott, K.; Klingauf, J.; Edwardson, J.M.; Chirgadze, D.Y.; Jackson, A.P. Crystal structure and molecular imaging of the Nav channel β 3 subunit indicates a trimeric assembly. *J. Boil. Chem.* **2014**, *289*, 10797–811, doi:10.1074/jbc.M113.527994.
- Namadurai, S.; Yereddi, N.R.; Cusdin, F.S.; Huang, C.L.-H.; Chirgadze, D.Y.; Jackson, A.P. A new look at sodium channel β subunits. *Open Boil.* **2015**, *5*, 140192, doi:10.1098/rsob.140192.
- O'Malley, H.A.; Isom, L.L. Sodium channel β subunits: emerging targets in channelopathies. *Annu. Rev. Physiol.* **2015**, *77*, 481–504, doi:10.1146/annurev-physiol-021014-071846.
- Ogata, N.; Ohishi, Y. Molecular diversity of structure and function of the voltage-gated Na⁺ channels. *Jpn. J. Pharmacol.* **2002**, *88*, 365–377, doi:10.1254/jjp.88.365.
- Pan, X.; Li, Z.; Huang, X.; Huang, G.; Gao, S.; Shen, H.; Liu, L.; Lei, J.; Yan, N. Molecular basis for pore blockade of human Na⁺ channel Nav1.2 by the μ -conotoxin KIIIa. *Sci.* **2019**, *363*, 1309–1313, doi:10.1126/science.aaw2999.
- Pan, X.; Li, Z.; Zhou, Q.; Shen, H.; Wu, K.; Huang, X.; Chen, J.; Zhang, J.; Zhu, X.; Lei, J.; et al. Structure of the human voltage-gated sodium channel Nav1.4 in complex with β 1. *Sci.* **2018**, *362*, eaau2486, doi:10.1126/science.aau2486.
- Patino, G.A.; Isom, L.L. Electrophysiology and beyond: Multiple roles of Na⁺ channel β subunits in development and disease. *Neurosci. Lett.* **2010**, *486*, 53–59, doi:10.1016/j.neulet.2010.06.050.

- Payandeh, J.; Scheuer, T.; Zheng, N.; Catterall, W.A. The crystal structure of a voltage-gated sodium channel. *Nat.* **2011**, *475*, 353–358, doi:10.1038/nature10238.
- Pedretti, A.; Villa, L.; Vistoli, G. VEGA--an open platform to develop chemo-bio-informatics applications, using plug-in architecture and script programming. *J. Comput. Mol. Des.* **2004**, *18*, 167–173, doi:10.1023/b:jcam.0000035186.90683.f2.
- Pettersen, E.F.; Goddard, T.D.; Huang, C.C.; Couch, G.S.; Greenblatt, D.M.; Meng, E.C.; Ferrin, T.E. UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem* **2004**, *25*, 1605–1612.
- Ragsdale, D.; McPhee, J.; Scheuer, T.; Catterall, W. Molecular determinants of state-dependent block of Na⁺ channels by local anesthetics. *Sci.* **1994**, *265*, 1724–1728, doi:10.1126/science.8085162.
- Richards, F.M. Areas, volumes, packing and protein structure. *Annu. Rev. Biophys. Bioeng.* **1977**, *6*, 151–176, doi:10.1146/annurev.bb.06.060177.001055.
- Rubinstein, M.; Patowary, A.; Stanaway, I.B.; Mccord, E.; Nesbitt, R.R.; Archer, M.; Scheuer, T.; Nickerson, D.; Raskind, W.H.; Wijsman, E.M.; et al. Association of rare missense variants in the second intracellular loop of NaV1.7 sodium channels with familial autism. *Mol. Psychiatry* **2016**, *23*, 231–239, doi:10.1038/mp.2016.222.
- Ruiz, M.D.L.; Kraus, R.L. Voltage-Gated Sodium Channels: Structure, Function, Pharmacology, and Clinical Indications. *J. Med. Chem.* **2015**, *58*, 7093–7118, doi:10.1021/jm501981g.
- Russell, R.B.; Sasieni, P.D.; Sternberg, M.J. Supersites within superfolds. Binding site similarity in the absence of homology 1 Edited by J. Thornton. *J. Mol. Boil.* **1998**, *282*, 903–918, doi:10.1006/jmbi.1998.2043.
- Salvage C.S.; Huang, C.L.H.; Jackson A.P. Cell-Adhesion Properties of β -Subunits in the Regulation of Cardiomyocyte Sodium Channels. *Biomolecules* **2020**, *10*, 7, doi: 10.3390/biom10070989.
- Sánchez-Solano, A.; Islas, A.A.; Scior, T.; Paiz-Candia, B.; Millan-Perezpeña, L.; Salinas-Stefanon, E.M. Characterization of specific allosteric effects of the Na⁺ channel β 1 subunit on the Nav1.4 isoform. *Eur. Biophys. J.* **2016**, *46*, 485–494, doi:10.1007/s00249-016-1193-3.
- Sarvagalla, S.; Cheung, C.H.A.; Tsai, J.Y.; Hsiehd, H.P.; Coumar, M.S. Disruption of protein-protein interactions: Hot spot detection, structure-based virtual screening and in vitro testing for the anti-cancer drug target-surviving. *Rsc. Adv.* **2016**, *6*, 31947–31959.
- Scior, T.; Paiz-Candia, B.; Islas, Ángel A.; Sánchez-Solano, A.; Peña, L.M.-P.; Mancilla-Simbro, C.; Salinas-Stefanon, E.M. Predicting a double mutant in the twilight zone of low homology modeling for the skeletal muscle voltage-gated sodium channel subunit beta-1 (Nav1.4 β 1). *Comput. Struct. Biotechnol. J.* **2015**, *13*, 229–240, doi:10.1016/j.csbj.2015.03.005.
- Scouras, A.D.; Daggett, V. The dynamoics rotamer library: Amino acid side chain conformations and dynamics from comprehensive molecular dynamics simulations in water. *Protein Sci.* **2011**, *20*, 341–352, doi:10.1002/pro.565.
- Shapovalov, M.V.; Dunbrack, R.L. A Smoothed Backbone-Dependent Rotamer Library for Proteins Derived from Adaptive Kernel Density Estimates and Regressions. *Struct.* **2011**, *19*, 844–858, doi:10.1016/j.str.2011.03.019.
- Sheinerman, F.B.; Norel, R.; Honig, B. Electrostatic aspects of protein–protein interactions. *Curr. Opin. Struct. Boil.* **2000**, *10*, 153–159, doi:10.1016/s0959-440x(00)00065-8.
- Shen, H.; Liu, D.; Wu, K.; Lei, J.; Yan, N. Structures of human Nav1.7 channel in complex with auxiliary subunits and animal toxins. *Sci.* **2019**, *363*, 1303–1308, doi:10.1126/science.aaw2493.
- Shen, H.; Zhou, Q.; Pan, X.; Li, Z.; Wu, J.; Yan, N. Structure of a eukaryotic voltage-gated sodium channel at near-atomic resolution. *Sci.* **2017**, *355*, eaal4326, doi:10.1126/science.aal4326.
- Sousounis, K.; E Haney, C.; Cao, J.; Sunchu, B.; Tsonis, P.A. Conservation of the three-dimensional structure in non-homologous or unrelated proteins. *Hum. Genom.* **2012**, *6*, 10, doi:10.1186/1479-7364-6-10.
- Spitzer, R.; Cleves, A.E.; Varela, R.; Jain, A.N. Protein function annotation by local binding site surface similarity. *Proteins: Struct. Funct. Bioinform.* **2013**, *82*, 679–694, doi:10.1002/prot.24450.
- Starr, T.; Thornton, J.W. Epistasis in protein evolution. *Protein Sci.* **2016**, *25*, 1204–1218, doi:10.1002/pro.2897.

- Stevens, M.; Peigneur, S.; Tytgat, J. Neurotoxins and Their Binding Areas on Voltage-Gated Sodium Channels. *Front. Pharmacol.* **2011**, *2*, 71, doi:10.3389/fphar.2011.00071.
- Vargas, E.; Yarov-Yarovoy, V.; Khalili-Araghi, F.; Catterall, W.A.; Klein, M.L.; Tarek, M.; Lindahl, E.; Schulten, K.; Perozo, E.; Bezanilla, F.; et al. An emerging consensus on voltage-dependent gating from computational modeling and molecular dynamics simulations. *J. Gen. Physiol.* **2012**, *140*, 587–594, doi:10.1085/jgp.201210873.
- Vizcarra, C.L.; Mayo, S.L. Electrostatics in computational protein design. *Curr. Opin. Chem. Boil.* **2005**, *9*, 622–626, doi:10.1016/j.cbpa.2005.10.014.
- Waxman, S.G. Sodium channels, the electrogenosome and the electrogenistat: lessons and questions from the clinic. *J. Physiol.* **2012**, *590*, 2601–2612, doi:10.1113/jphysiol.2012.228460.
- Webb, B.; Sali, A. Protein Structure Modeling with MODELLER. *Advanced Structural Safety Studies*. **2014**, *1137*, 1–15, doi:10.1007/978-1-4939-0366-5_1.
- Whisstock, J.; Lesk, A. Prediction of protein function from protein sequence and structure. *Q. Rev. Biophys.* **2003**, *36*, 307–340, doi:10.1017/s0033583503003901.
- Woolf, C.J.; Mannion, R.J. Neuropathic pain: aetiology, symptoms, mechanisms, and management. *Lancet* **1999**, *353*, 1959–1964, doi:10.1016/s0140-6736(99)01307-0.
- Yan, Z.; Zhou, Q.; Wang, L.; Wu, J.; Zhao, Y.; Huang, G.; Peng, W.; Shen, H.; Lei, J.; Yan, N. Structure of the Na^v 1.4- β 1 Complex from Electric Eel. *Cell* **2017**, *170*, 470–482.e11, doi:10.1016/j.cell.2017.06.039.
- Yerreddi, N.R.; Cusdin, F.S.; Namadurai, S.; Packman, L.C.; Monie, T.P.; Slavny, P.; Clare, J.J.; Powell, A.J.; Jackson, A.P. The immunoglobulin domain of the sodium channel β 3 subunit contains a surface-localized disulfide bond that is required for homophilic binding. *FASEB J.* **2012**, *27*, 568–580, doi:10.1096/fj.12-209445.
- Yu, E.J.; Ko, S.-H.; Lenkowski, P.W.; Pance, A.; Patel, M.K.; Jackson, A.P. Distinct domains of the sodium channel β 3-subunit modulate channel-gating kinetics and subcellular location. *Biochem. J.* **2005**, *392*, 519–526, doi:10.1042/bj20050518.
- Yu, F.; Catterall, W. The VGL-Chanome: A Protein Superfamily Specialized for Electrical Signaling and Ionic Homeostasis. *Science's STKE*, **2004**, re15.
- Yu, F.H.; Westenbroek, R.E.; Silos-Santiago, I.; McCormick, K.A.; Lawson, D.; Ge, P.; Ferriera, H.; Lilly, J.; Distefano, P.S.; Catterall, W.A.; et al. Sodium Channel β 4, a New Disulfide-Linked Auxiliary Subunit with Similarity to β 2. *J. Neurosci.* **2003**, *23*, 7577–7585, doi:10.1523/JNEUROSCI.23-20-07577.2003.
- Yu, F.H.; Yarov-Yarovoy, V.; Gutman, G.A.; Catterall, W.A. Overview of Molecular Relationships in the Voltage-Gated Ion Channel Superfamily. *Pharmacol. Rev.* **2005**, *57*, 387–395, doi:10.1124/pr.57.4.13.
- Zahiri, J.; Bozorgmehr, J.H.; Masoudi-Nejad, A. Computational Prediction of Protein-Protein Interaction Networks: Algorithms and Resources. *Curr. Genom.* **2013**, *14*, 397–414, doi:10.2174/1389202911314060004.
- Zhu, W.; Voelker, T.L.; Varga, Z.; Schubert, A.R.; Nerbonne, J.M.; Silva, J. Mechanisms of noncovalent β subunit regulation of Nav channel gating. *J. Gen. Physiol.* **2017**, *149*, 813–831, doi:10.1085/jgp.201711802.

12. MATERIAL SUPLEMENTARIO

El Material Suplementario fue necesario para documentar los hallazgos esenciales como una sinopsis tomada de un número aún mayor de estudios quimiométricos realizados sistemáticamente sobre los canales de sodio:

1. Alineamientos de secuencia múltiple (MSA): 9 (isoformas) x 3 (especies) x 16 (ECL) = 432 secuencias primarias.
2. Modelos 3D de estructuras molde determinadas por cryo-EM: 9 x 3 x 2 (β_1, β_3) = 54 archivos PDB.
3. Cálculos de superficies: 9 x 3 x 16 = 432 modelos de superficie x 2 ($\alpha+\beta$) = 864 superficies
4. Cálculos de superficies polares y no polares: (9 x 3 x 16) x 2 (PSA, NPSA) = 432 x 2 propiedades.
5. Cálculos de volúmenes moleculares: 9 x 3 x 16 = 432 patrones de volumen.
6. Cálculo del área de interfaz (área enterrada): 9 x 3 x 2 (α/β) x 4 (ECL) = 216 números de regiones enterradas entre las subunidades α y β .
7. Cálculos del área IF de nueve isoformas por tres especies por cuatro IF-ECL: 9 x 3 x 2 (β_1, β_3) x 4 (IF-ECL) x 2 (α / β) = 432 número total de patrones de superficie.
8. Cálculos del área IF de los átomos polares y apolares que forman la interfaz: 9 x 3 x 2 (β_1, β_3) x 4 (IF-ECLs) x 2 (α / β) = 432 patrones atómicos polares y 432 no polares.
9. Cálculos del área MEPS (rojo = negativo, azul = positivo) basados en los vértices de la superficie accesible al solvente de los átomos que componen la interfaz: 9 x 3 x 2 (β_1, β_3) x 4 (IF- ECL) x 2 (α / β) = 432 patrones de carga positiva y 432 patrones de carga negativa.
10. Recuento de las longitudes de las asas: 9 (isoformas) x 3 (especies) x 16 (ECL) = 432 recuentos.
11. Recuento de residuos polares: 9 (isoformas) x 3 (especies) x 16 (ECL) = 432 recuentos.
12. Recuento de los residuos de cisteína como formación potencial de puentes S-S: 9 (isoformas) x 3 (especies) x 16 (ecl) = 432 recuentos.
13. Recuento de residuos aromáticos cargados positiva o negativamente: 9 (isoformas) x 3 (especies) x 16 (ECL) x 3 = 432 x 3 recuentos.

Tabla S1. PPIs de los residuos de interfaz de Nav α /Nav β

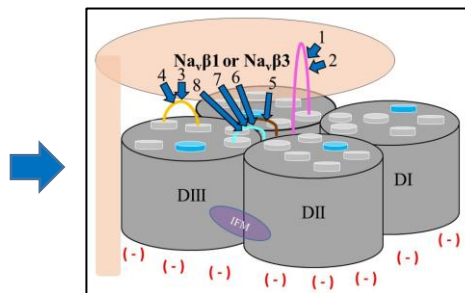
Nav α	ECL PPI Id ³	S5 DI		S1-S2 DIII		S5 DIV		S6 DIV	
		1	2	3	4	5	6	7	8
hNav.1.1 ⁷	P35498 ²	agq C peg(O)	qcp E gym(O)	yid Q rkt(N)	dqr K tik(N)	gid D mfN(O)	Pnk V npg ^{NI}	nkV N pgs(O)	kvn P gss(O)
mNav.1.1 ⁷	A2APX8 ²	agq C peg(O)	qcp E gym(O)	yid Q rkt(N)	dqr K tik(N)	gid D mfN(O)	pnk V npg ^{NI}	nkV N pgs(O)	kvn P gss(O)
rNav.1.1 ⁷	P04774 ²	agq C peg(O)	qcp E gym(O)	yid Q rkt(N)	dqr K tik(N)	gid D mfN(O)	pnk V npg ^{NI}	nkV N pgs(O)	kvn P gss(O)
hNav.1.2 ⁵	6J8E ¹	agq C peg(O)	qcp E gyi(O)	yie Q rkt(N)	eqr K tik(N)	gid D mfN(O)	pdK D hpg(O)	dkd H pgs(N)	kdh P gss(O)
mNav.1.2 ⁷	B1AWN6	agq C peg(O)	qcp E gyi(O)	yie Q rkt(N)	eqr K tik(N)	gid D mfN(O)	pek D hpg(O)	ekd H pgs(N)	kdh P gss(O)
rNav.1.2 ⁷	P04775 ²	agq C peg(O)	qcp E gyi(O)	yie Q rkt(N)	eqr K tik(N)	gid D mfN(O)	pek D hpg(O)	ekd H pgs(N)	kdh P gss(O)
hNav.1.3 ⁷	Q9NY46 ²	agq C peg(O)	qcp E gyi(O)	yie Q rkt(N)	eqr K tik(N)	gid D mfN(O)	pdT I hpg ^{NI}	dti H pgs(N)	tih P gss(O)
mNav.1.3 ⁷	A2ASI5 ²	agq C peg(O)	qcp E gyi(O)	yie Q rkt(N)	eqr K tik(N)	gid D mfN(O)	pda I hpg ^{NI}	dai H pgs(N)	aih P gss(O)
rNav.1.3 ⁷	P08104 ²	agq C peg(O)	qcp E gyi(O)	yie Q rkt(N)	eqr K tik(N)	gid D mfN(O)	pda I hpg ^{NI}	dai H pgs(N)	aih P gss(O)
eeNav.1.4 ⁵	5XSY ¹	agk C peg ^{NI}	kcp E gyt(O)	yiW R rrv(N)	wrr R vik(N)	gvd D ifN(O)	pdV E npG(O)	dve N pgt(O)	ven P gtd(O)
hNav.1.4 ⁵	6AGF ¹	agh C peg(O)	hcp E gye(O)	yie Q rrv(N)	eqr R vir(N)	gid D mfN(O)	pnL E npG(O)	nle N pgt(O)	len P gts(O)
mNav.1.4 ⁷	Q9ER60 ²	agh C peg(O)	hcp E gye(O)	yie Q rrv(N)	eqr R vir(N)	gid D mfN(O)	ptL E npG(O)	tle N pgt(O)	len P gtn(O)
rNav.1.4 ⁷	P15390 ²	agh C peg(O)	hcp E gye(O)	yie Q rrv(N)	eqr R vir(N)	gid D mfN(O)	ptL E npG(O)	tle N pgt(O)	len P gtn(O)
hNav.1.5 ⁷	Q14524 ²	agt C peg(O)	tcp E gyr(O)	yle E rkt ^{NI}	eer K tik(N)	gid D mfN(O)	ptL P nsn ^{NI}	tlp N sng(O)	lpn S ngs(O)
mNav.1.5 ⁷	Q9JJV9 ²	agt C peg(O)	tcp E gyr(O)	yle E rkt ^{NI}	eer K tik(N)	gid D mfN(O)	pnL P nsn ^{NI}	nlp N sng(O)	lpn S ngs(O)
rNav.1.5 ⁵	6U70 ¹	agt C peg(O)	tcp E gyr(O)	yle E rkt ^{NI}	eer K tik(N)	gid D mfN(O)	pnL P nsn ^{NI}	nlp N sng(O)	lpn S ngs(O)
hNav.1.6 ⁷	Q9UQD0	agq C peg(O)	qcp E gyq(O)	yie Q rkt(N)	eqr K tir(N)	gid D mfN(O)	ldk E hpg(O)	dke H pgs(N)	keh P gsg(O)
mNav.1.6 ⁷	Q9WTU3	agq C peg(O)	qcp E gfq(O)	yie Q rkt(N)	eqr K tir(N)	gid D mfN(O)	ldk E hpg(O)	dke H pgs(N)	keh P gsg(O)
rNav.1.6 ⁷	O88420 ²	agq C peg(O)	qcp E gfq(O)	yie Q rkt(N)	eqr K tir(N)	gid D mfN(O)	ldk E hpg(O)	dke H pgs(N)	keh P gsg(O)
hNav.1.7 ⁵	6J8G ¹	sgq C peg(O)	qcp E gyt(O)	yie R kkk(N)	erk K tik(N)	gin D mfN(O)	pkk V hpg ^{NI}	kkv H pgs(N)	kvh P gss(O)
mNav.1.7 ⁷	Q62205 ²	sgq C peg(O)	qcp E gye(O)	yie K kkk(N)	ekk K tik(N)	gin D mfN(O)	pkk V hpg ^{NI}	kkv H pgs(N)	kvh P gss(O)
rNav.1.7 ⁷	O08562 ²	sgq C peg(O)	qcp E gyi(O)	yie K kkk(N)	ekk K tik(N)	gin D mfN(O)	pkk V hpg ^{NI}	kkv H pgs(N)	kvh P gss(O)
hNav.1.8 ⁷	Q9Y5Y9 ²	sgH C pdg(O)	hcp D gyi(O)	ylD Q kpt(N)	dqk P tvk ^{NI}	gid D mfN(O)	pnL P nsn ^{NI}	nlp N sng(O)	lpn S ngt(O)
mNav.1.8 ⁷	Q6QIY3 ²	agh C pnd(O)	hcp N dyv(O)	yle E kpr ^{NI}	eek P rvk ^{NI}	gid D mfN(O)	pnR P nsn ^{NI}	nrp N sng(O)	rpn S ngs(O)
rNav.1.8 ⁷	Q62968 ²	agh C pgg(O)	hcp G gyv ^{NI}	yle E kpr ^{NI}	eek P rvk ^{NI}	gid D mfN(O)	pnL P nsn ^{NI}	nlp N sng(O)	lpn S ngs(O)
hNav.1.9 ⁷	Q9UI33 ²	nsa C siq(O)	acs I qye ^{NI}	hle N qpk(N)	enq P kiqu ^{NI}	gid D ifN(O)	rsk E scn ^{NI}	ske S cns ^{NI}	kes C ns ^{NI}
mNav.1.9 ⁷	Q9R053 ²	rrs C pdg(O)	scp D gst(O)	nlp S rpq ^{NI}	psr P qve ^{NI}	gid D ifN(O)	esk A scn ^{NI}	ska S cns ^{NI}	kas C ns ^{NI}
rNav.1.9 ⁷	O88457 ²	srp C png(O)	pcp N gst(O)	nlp S rpq ^{NI}	psr P qve ^{NI}	gid D ifN(O)	eak E hcn ^{NI}	ake H cns ^{NI}	keh C ns ^{NI}
Nav β	PPI Id ⁴	1b	2b	3b	4b	5b	6b	7b	8b
eeNav.1.1 ⁵	5XSY ¹	sck M rge ^{NI}	yfd R tlf(N)	acv E vds(O)	vds D tea(O)	ckm R gev(N)	mgs K ntf(N)	sng A cve(N)	gac V evd(N)
hNav.1.1 ⁵	6AGF ¹	sck R rse(N)	hvy R llf(N)	gcv E vds(O)	vds E tea(O)	ckr R set(N)	ngs R gtk(N)	acg G cve(N)	ggc V evd(N)
mNav.1.1 ⁷	P97952 ²	sck R rse(N)	hvy R llf(N)	gcv E vds(O)	vds D tea(O)	ckr R set(N)	ngs R gtk(N)	awg G cve(N)	ggc V evd(N)
rNav.1.1 ⁷	Q00954 ²	sck R rse(N)	hvy R llf(N)	gcv E vds(O)	vds E tea(O)	ckr R set(N)	ngs R gtk(N)	awg G cve(N)	ggc V evd(N)
hNav.1.3 ⁶	4L1D ¹	scm K ree(N)	nvs R efe(N)	vcv E vps(O)	vps E tea(O)	cmk R eev(N)	ngs K dlq(N)	cfp V cve(N)	pvc V evp(N)
mNav.1.3 ⁷	Q8BHK2	scm K ree(N)	nvs R efe(N)	vcv E vps(O)	vps E tea(O)	cmk R eev(N)	ngs K dlq(N)	cfp V cve(N)	pvc V evp(N)
rNav.1.3 ⁷	Q9JK00 ²	scm K ree(N)	nvs R efe(N)	vcv E vps(O)	vps E tea(O)	cmk R eev(N)	ngs K dlq(N)	cfp V cve(N)	pvc V evp(N)

¹Código PDB (<http://www.rcsb.org/>), ²código UniProt (<https://www.uniprot.org/>), ³PPI-Id para las posiciones de las interacciones polares calculadas los residuos en Nav α (1–8), ⁴PPI-Id para las posiciones de las interacciones polares calculadas de los residuos en Nav β 1 y Nav β 3 (1b–8b) con respecto a Nav α ; estructuras moldes (⁵Cryo-EM, ⁶estructuras determinadas por cristalografía de rayos X), ⁷modelos, NI: no interacción; residuos que forman interacciones polares calculadas (letras mayúsculas en negra); aminoácidos adyacentes en secuencia a residuos que forman interacciones polares calculadas (letras minúsculas); residuos cargados positivos y negativos (azul y rojo, respectivamente), residuos polares (cian), residuos no polares (naranja).

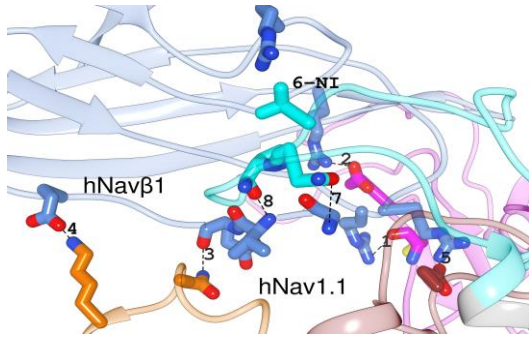
TABLA S1 (Alternativa)

Presentación alternativa a las Tablas S1 y 3

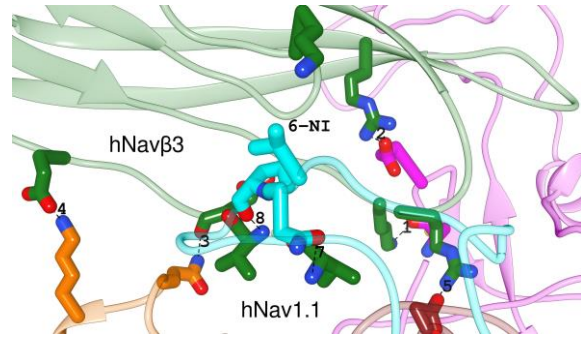
Isoform	Codes of PDB or UniProt entries	Extracellular loop							
		S5 DI	S1-S2 DIII		S5 DIV	S6 DIV			
		Id of interactions							
		1	2	3	4	5	6	7	8
*eeNav 1.4	5XSY ^{PDB}	NO	YES	YES	YES	YES	YES	YES	YES
hNav 1.1	P35498A ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
mNav 1.1	A2APX8	YES	YES	YES	YES	YES	NO	YES	YES
rNav 1.1	P04774 ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
hNav 1.2	6J8E ^{PDB}	YES	YES	YES	YES	YES	YES	YES	YES
mNav 1.2	A0A0R5R P28 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
rNav 1.2	P04775 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
hNav 1.3	Q9NY46 ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
mNav 1.3	A2AYES5 ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
rNav 1.3	P08104 ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
hNav 1.4	6AGF ^{PDB}	YES	YES	YES	YES	YES	YES	YES	YES
mNav 1.4	Q9ER60 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
rNav 1.4	P15390 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
hNav 1.5	Q14524 ^{UP}	YES	YES	NO	YES	YES	NO	YES	YES
mNav 1.5	Q9JJV9 ^{UP}	YES	YES	NO	YES	YES	NO	YES	YES
rNav 1.5	6U70 ^{PDB}	YES	YES	NO	YES	YES	NO	YES	YES
hNav 1.6	Q9UQD0 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
mNav 1.6	Q9WTU3 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
rNav 1.6	O88420 ^{UP}	YES	YES	YES	YES	YES	YES	YES	YES
hNav 1.7	6J8G ^{PDB}	YES	YES	YES	YES	YES	NO	YES	YES
mNav 1.7	Q62205 ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
rNav 1.7	O08562 ^{UP}	YES	YES	YES	YES	YES	NO	YES	YES
hNav 1.8	Q9Y5Y9 ^{UP}	YES	YES	YES	NO	YES	NO	YES	YES
mNav 1.8	Q6QIY3 ^{UP}	YES	YES	NO	NO	YES	NO	YES	YES
rNav 1.8	Q62968 ^{UP}	YES	NO	NO	NO	YES	NO	YES	YES
hNav 1.9	Q9UI33 ^{UP}	YES	NO	YES	NO	YES	NO	NO	NO
mNav 1.9	Q9R053 ^{UP}	YES	YES	NO	NO	YES	NO	NO	NO
rNav 1.9	O88457 ^{UP}	YES	YES	NO	NO	YES	NO	NO	NO



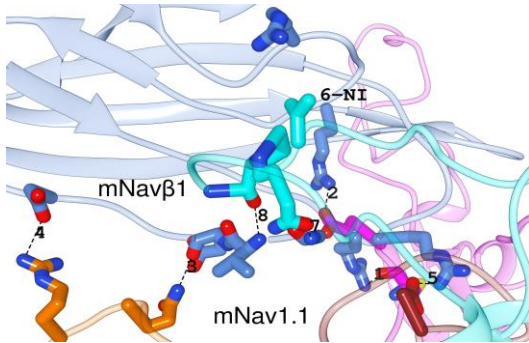
Azul = Puente de hidrógeno
Amarillo = Puente de sal
Rojo = No interacción



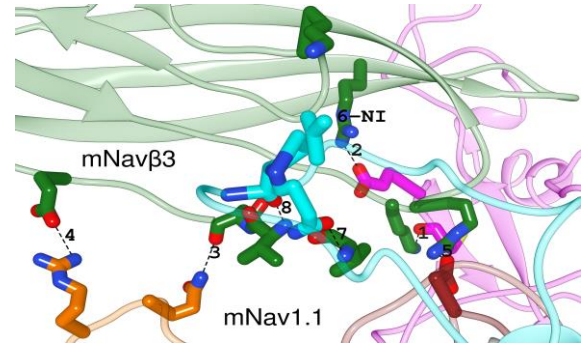
(a)



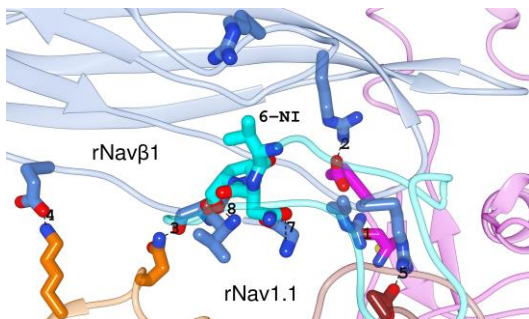
(b)



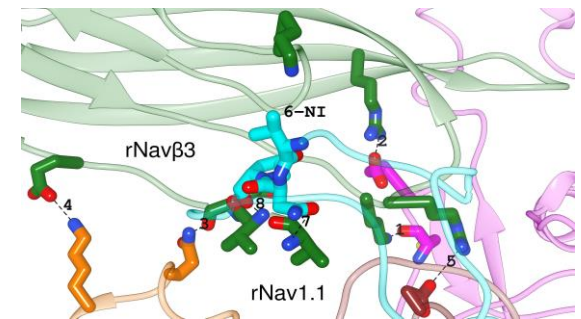
(c)



(d)



(e)



(f)

Figura S1. PPI de las tres isoformas h, m, rNav1.1 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

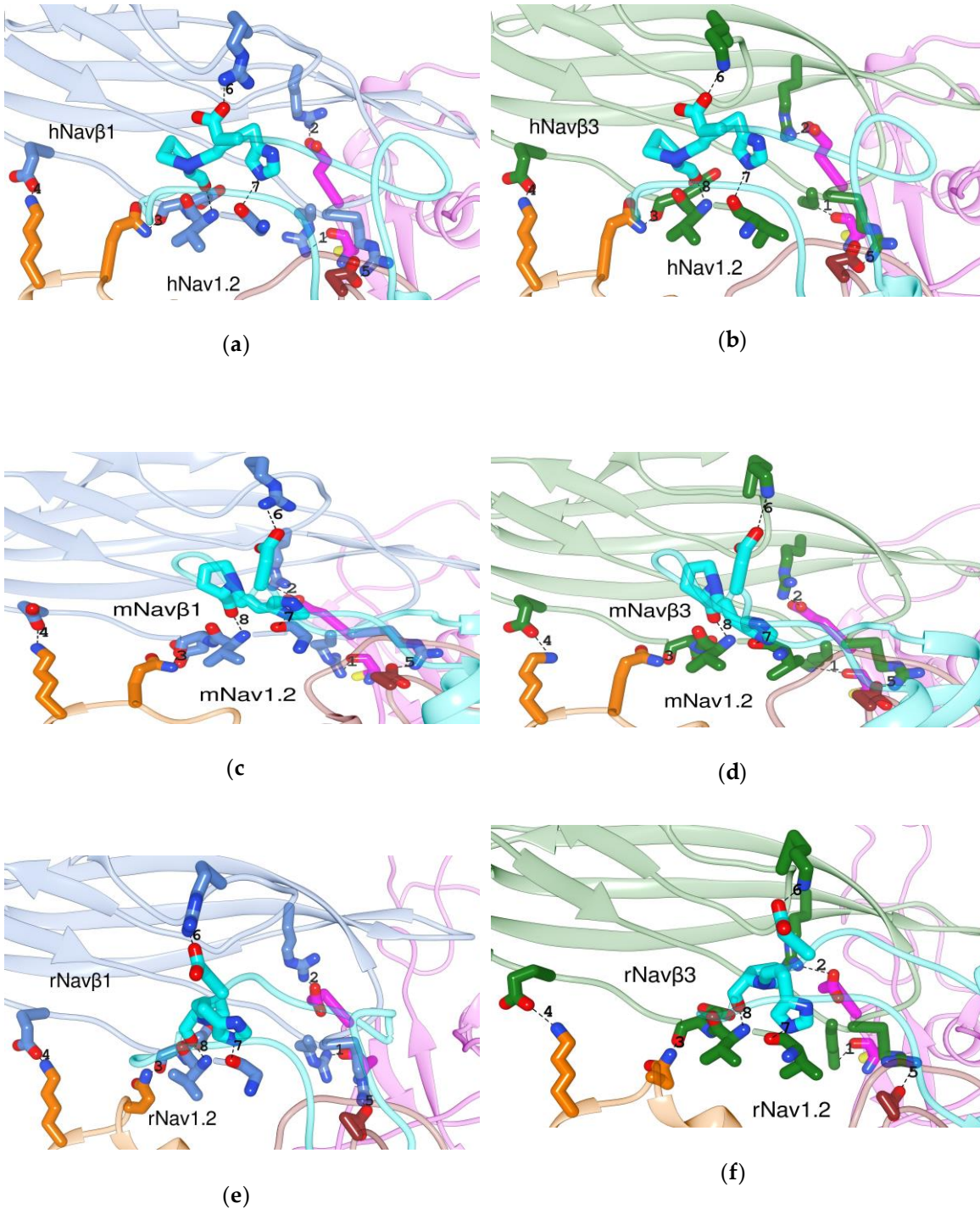


Figura S2. PPI de las tres isoformas h, m, rNav1.2 en interfaz con Nav β 1 y Nav β 3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1 -S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Nav β 1: azul cian; Nav β 3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

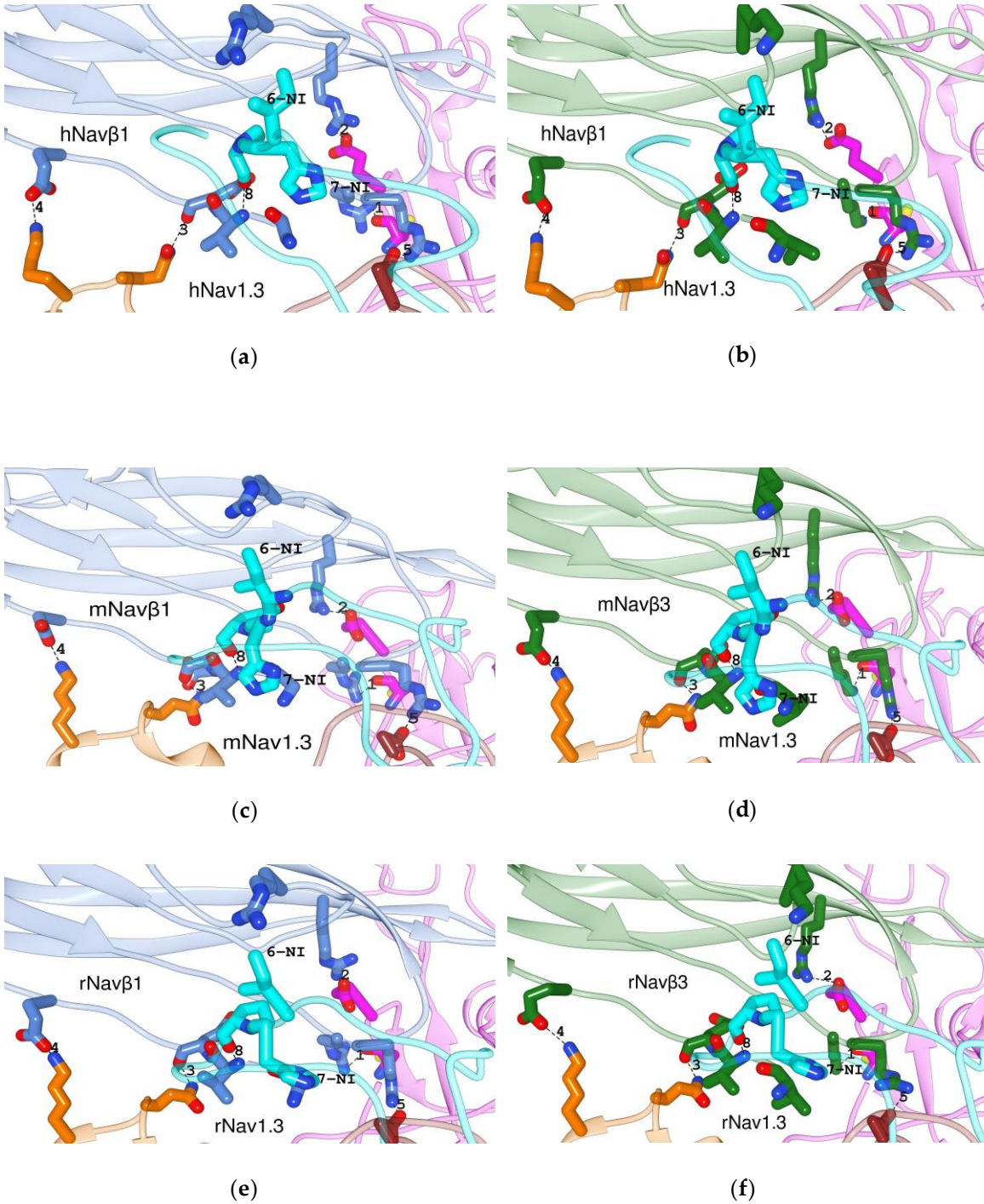


Figura S3. PPI de las tres isoformas h, m, rNav1.3 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

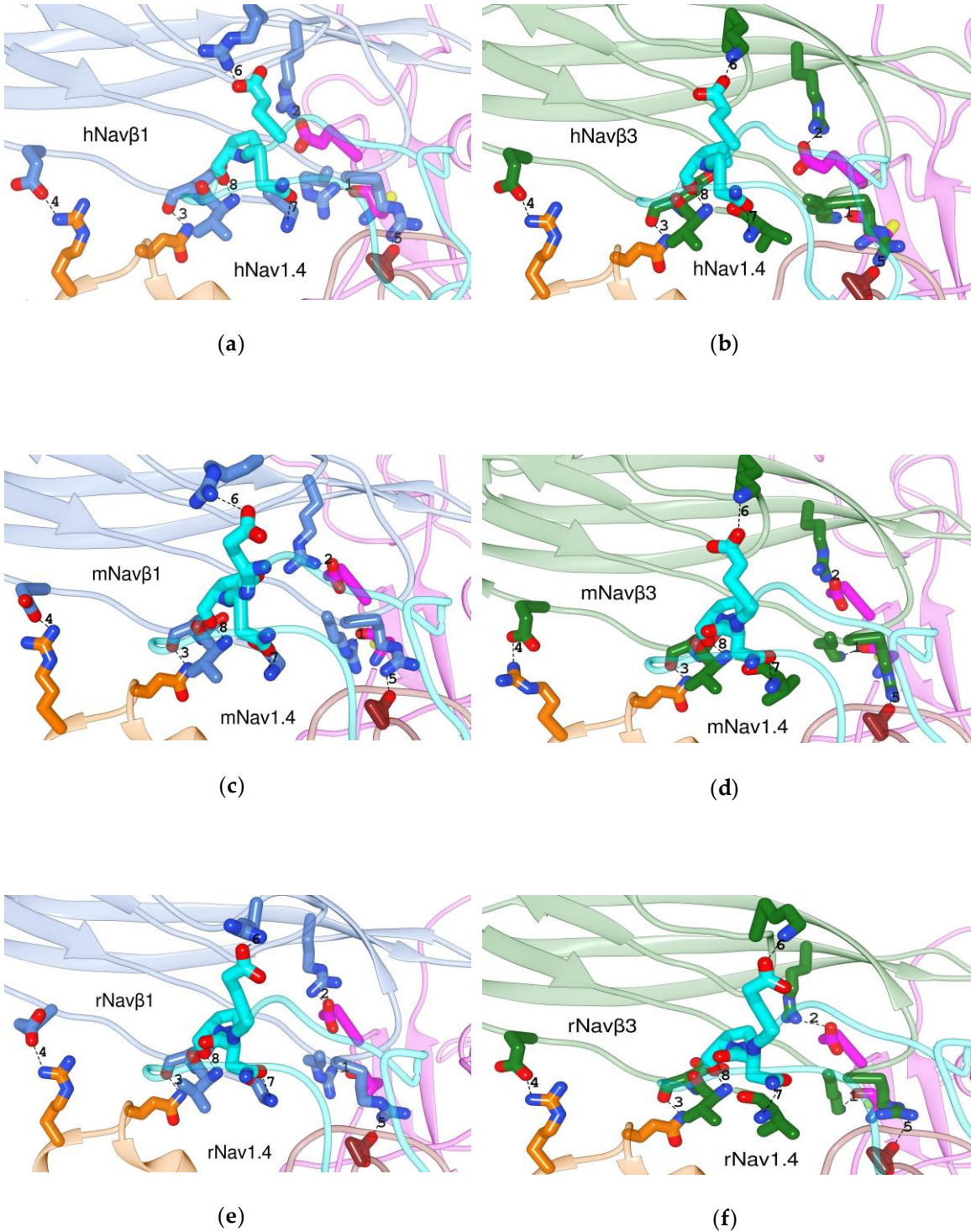


Figura S4. PPI de las tres isoformas h, m, rNav1.4 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DI: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

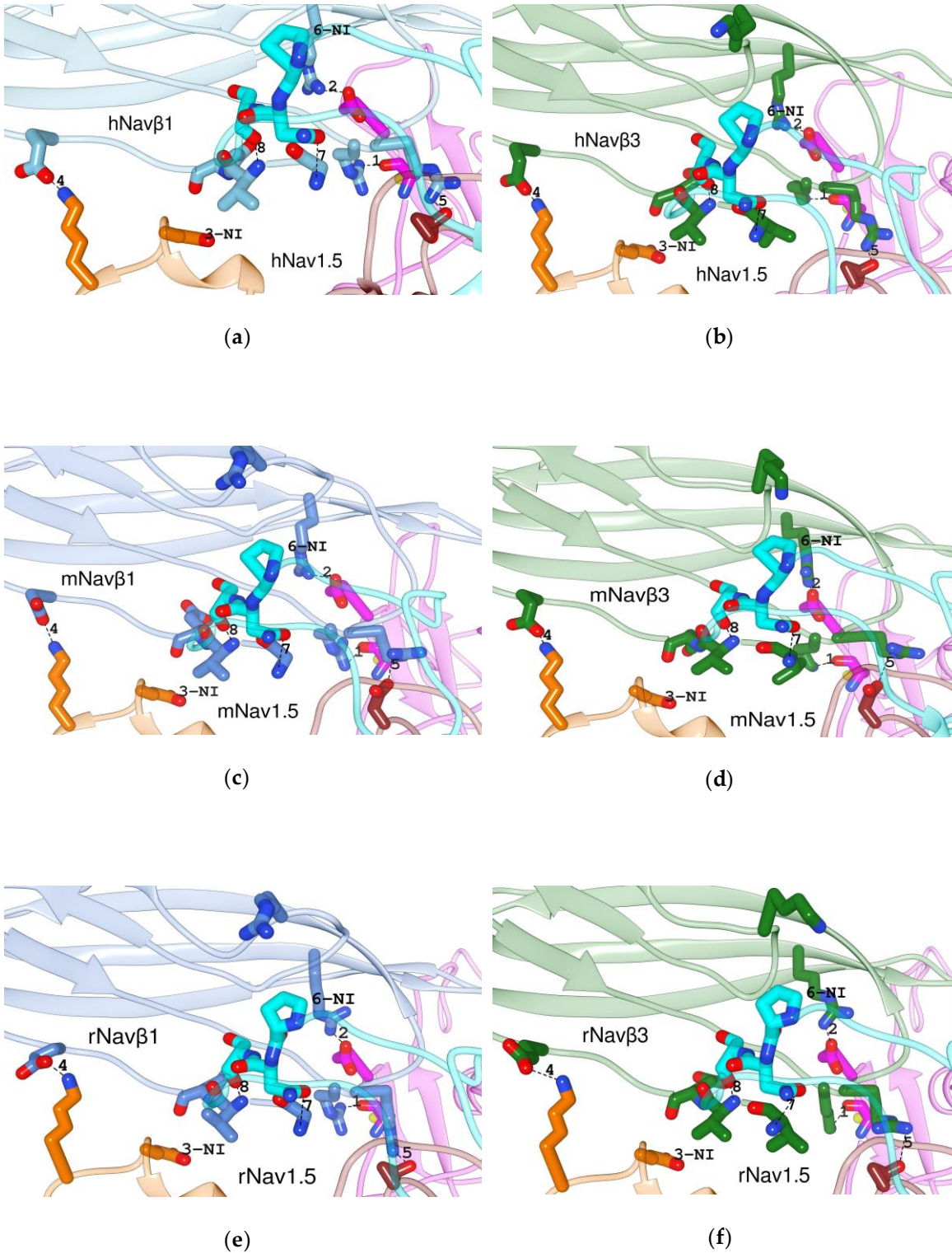


Figura S5. PPI de las tres isoformash, m, rNav1.5 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

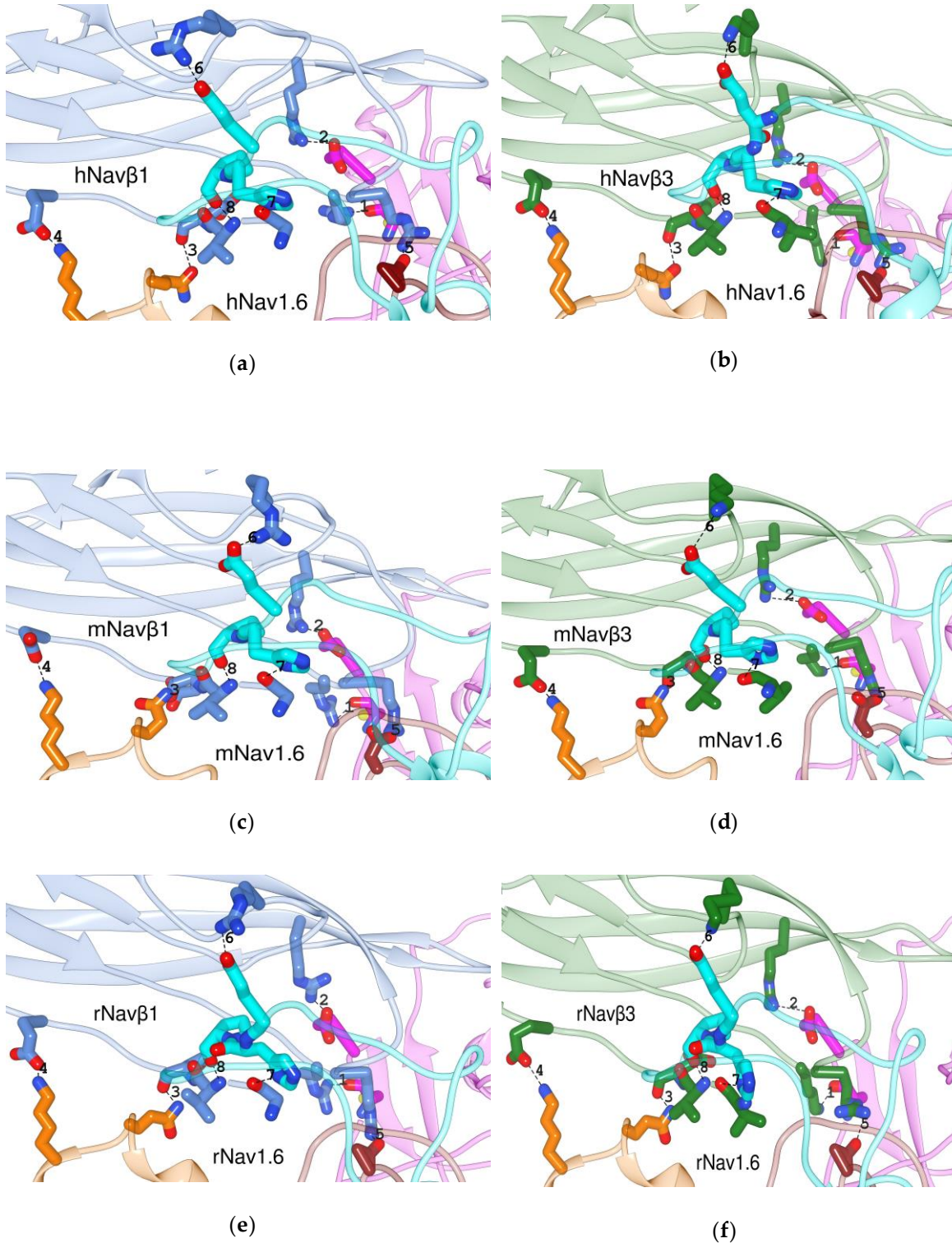


Figura S6. PPI de las tres isoformas h, m, rNav1.6 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

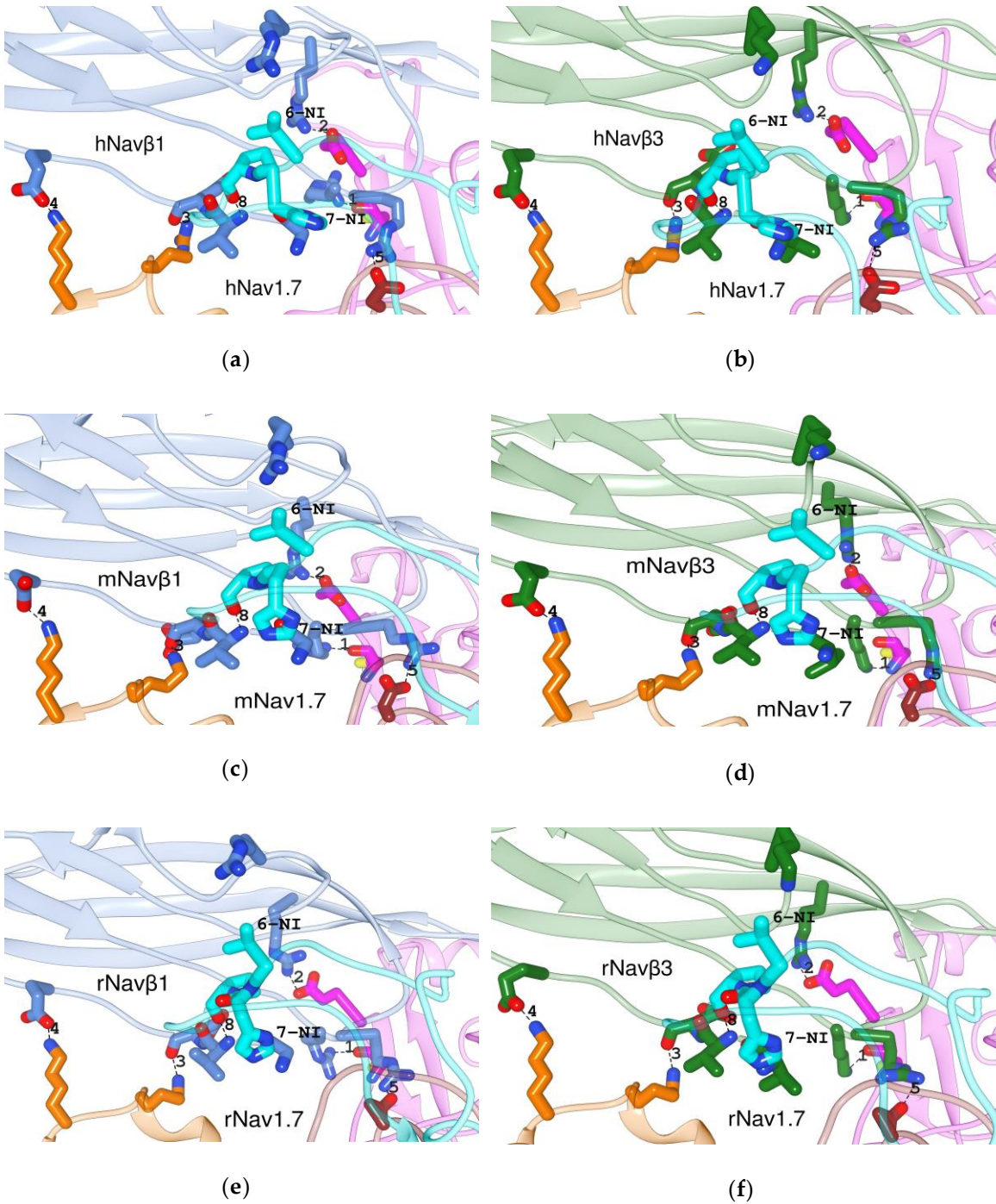


Figura S7. PPI de las tres isoformas h, m, rNav1.7 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

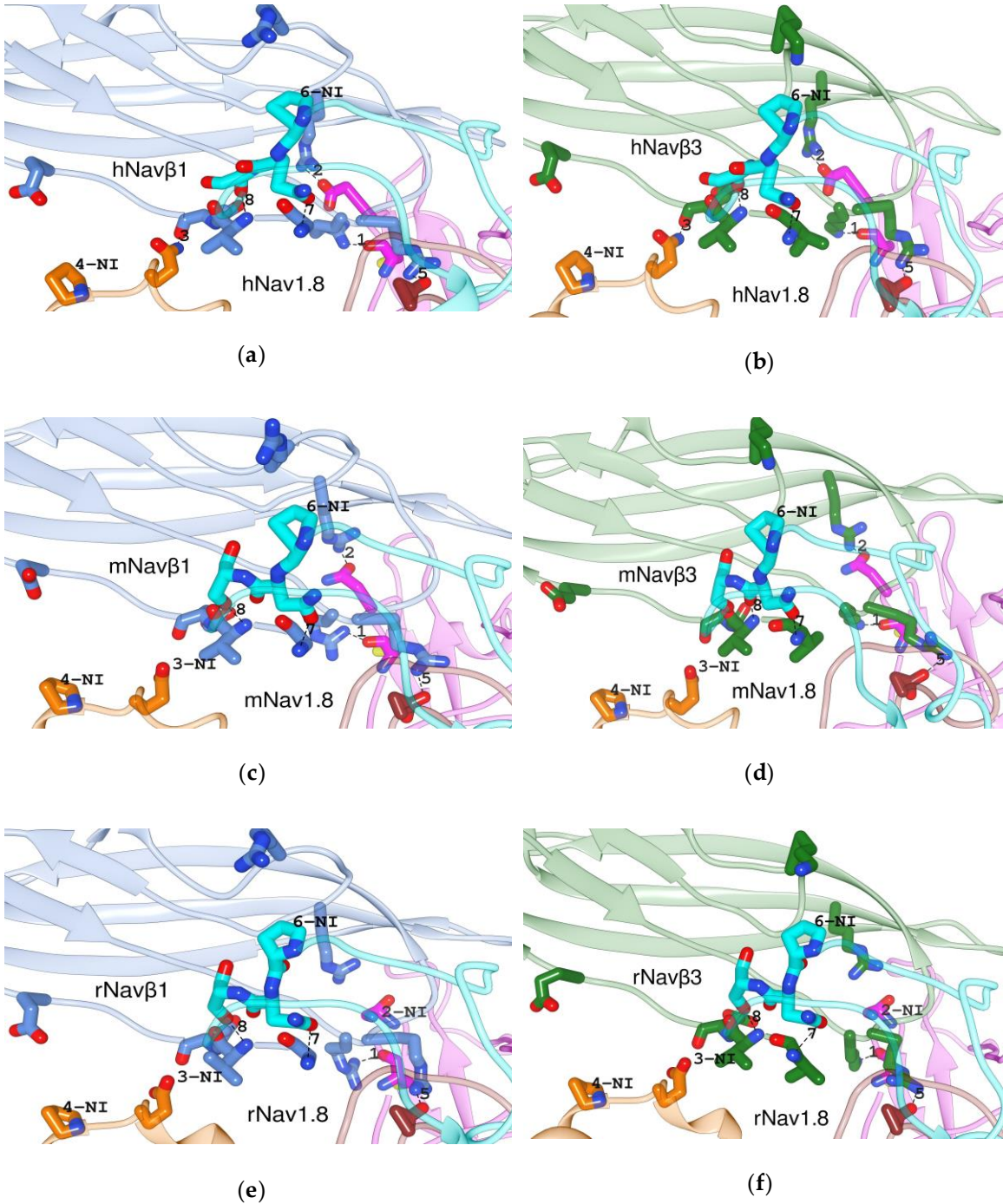


Figura S8. PPI de las tres isoformas h, m, rNav1.8 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

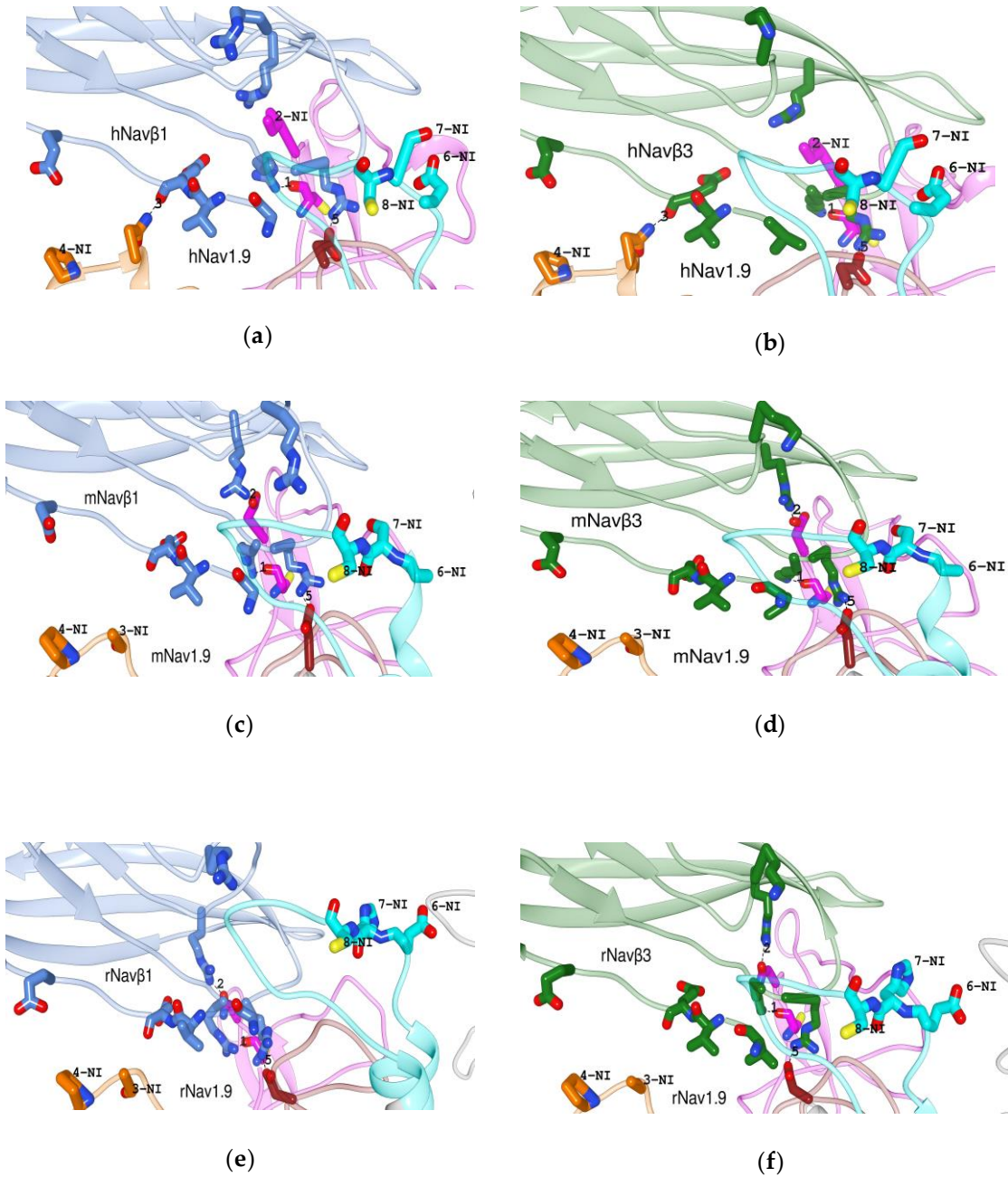


Figura S9. PPI de las tres isoformas h, m, rNav1.9 en interfaz con Navβ1 y Navβ3. (a, b) complejos de humano; (c, d) ratón; (e, f) rata; secciones que no participan en la interfaz: gris; S5 DI: magenta; S1-S2 DIII: naranja; S5 DIV: marrón; S6 DIV: cian; Navβ1: azul cian; Navβ3: verde bosque; números: PPI-Ids según la Tablas S1 y 3; NI: sin interacción; líneas punteadas: interacciones polares.

Tabla S2. Área de interfaz de las IF-ECLs/Navas.

Isoforma de Nav α	IF-ECLs de Nav α	Nav β 1	Nav β 3	Nav β 1	Nav β 3	Nav β 1	Nav β 3
Nav1.1 α	S5 DI	621.8	584.1	664.3	527.2	491.0	367.6
	S1-S2 DIII	294.4	339.8	308.4	299.6	263.0	332.6
	S5 DIV	120.2	151.9	163.1	170.8	137.1	164.6
	S6 DIV	217.5	264.1	270.0	295.9	210.0	327.5
Nav1.2 α	S5 DI	587.7	552.4	612.8	485.6	535.8	490.1
	S1-S2 DIII	333.8	360.1	350.4	327.3	317.6	380.6
	S5 DIV	136.7	174.5	171.4	158.9	140.5	162.4
	S6 DIV	208.8	254.6	234.0	259.2	220.8	357.5
Nav1.3 α	S5 DI	602.3	544.5	664.0	554.6	498.0	387.7
	S1-S2 DIII	293.1	330.3	313.2	309.4	315.0	378.9
	S5 DIV	137.9	183.6	171.8	162.6	130.7	135.3
	S6 DIV	187.1	251.5	249.4	272.4	215.4	312.2
Nav1.4 α	S5 DI	578.6	550.6	559.4	441.0	549.9	434.7
	S1-S2 DIII	304.8	341.4	359.2	350.4	318.0	399.1
	S5 DIV	144.0	170.6	154.6	135.4	129.0	158.5
	S6 DIV	248.2	268.2	284.9	282.3	227.2	300.1
Nav1.5 α	S5 DI	570.0	534.6	610.2	483.8	535.7	417.3
	S1-S2 DIII	324.7	360.3	368.6	362.0	293.9	357.9
	S5 DIV	175.3	200.0	189.5	184.6	170.9	197.0
	S6 DIV	249.3	281.2	280.7	274.9	218.7	314.3
Nav1.6 α	S5 DI	525.5	471.6	552.9	440.3	465.3	388.4
	S1-S2 DIII	325.5	360.3	294.8	278.5	287.4	334.5
	S5 DIV	147.2	169.5	167.4	162.1	119.5	145.0
	S6 DIV	199.2	253.7	269.3	288.1	216.1	314.0
Nav1.7 α	S5 DI	665.5	593.6	663.3	544.9	576.2	461.7
	S1-S2 DIII	381.9	425.8	389.6	380.2	376.9	428.6
	S5 DIV	108.5	141.4	143.9	140.4	89.0	122.7
	S6 DIV	203.5	236.8	232.6	252.1	183.5	261.1
Nav1.8 α	S5 DI	585.5	551.0	623.7	510.7	583.5	464.8
	S1-S2 DIII	292.5	343.5	279.5	273.7	261.0	330.9
	S5 DIV	179.3	207.6	165.8	136.8	115.7	135.3
	S6 DIV	125.0	162.7	262.0	260.1	211.3	309.4
Nav1.9 α	S5 DI	633.0	575.7	602.7	520.6	479.9	306.7
	S1-S2 DIII	317.7	385.5	266.2	271.1	298.4	368.2
	S5 DIV	141.6	186.1	138.8	133.1	96.4	128.7
	S6 DIV	17.7	24.9	14.7	25.7	30.0	34.9

Las unidades de área son en Å^2 . La superficie de interfaz de area enterrada fue calculada

Tabla S3. Validación de la secuencia de residuos de la interfaz entre la subunidad Nav α y las subunidades Nav β 1 y Nav β 3.

Finalmente, se documentó la zona de interacción proteína-proteína calculada por HOTREGION para la estructura molde de anguila para modelos 3D de canales de sodio (código PDB: 5XSY, [Yan *et al.*, 2017]) mediante una herramienta bioinformática basada en la web en <http://prism.cccb.ku.edu.tr/hotregion/>. Los residuos identificados corresponden a todos aquellos que también pueden ser estudiados por la herramienta Chimera. Para obtener más detalles, consulte la documentación del software. La lista de salida de residuos que interactúan entre ambas subunidades para el canal de sodio de la anguila (código PDB: 5XSY) por HotRegion es una base de datos de puntos calientes cooperativos.

Interface Name	Residue Number	Residue Type	Chain
5XSYBA	273	ARG	A
5XSYBA	301	GLU	A
5XSYBA	313	LEU	A
5XSYBA	323	LYS	A
5XSYBA	324	CYS	A
5XSYBA	325	PRO	A
5XSYBA	326	GLU	A
5XSYBA	327	GLY	A
5XSYBA	985	ILE	A
5XSYBA	988	ASN	A
5XSYBA	989	LEU	A
5XSYBA	992	THR	A
5XSYBA	996	ILE	A
5XSYBA	1005	PHE	A
5XSYBA	1022	ILE	A
5XSYBA	1023	TYR	A
5XSYBA	1025	TRP	A
5XSYBA	1026	ARG	A
5XSYBA	1028	ARG	A
5XSYBA	1032	VAL	A
5XSYBA	1036	TYR	A
5XSYBA	1039	LYS	A
5XSYBA	1040	VAL	A
5XSYBA	1043	TYR	A
5XSYBA	1044	VAL	A
5XSYBA	1047	VAL	A
5XSYBA	1475	TYR	A
5XSYBA	1484	ASP	A
5XSYBA	1489	GLU	A
5XSYBA	1528	ASN	A
5XSYBA	1529	PRO	A
5XSYBA	1530	GLY	A
5XSYBA	1531	THR	A
5XSYBA	24	ALA	B
5XSYBA	25	CYS	B
5XSYBA	26	VAL	B
5XSYBA	27	GLU	B

5XSYBA	28	VAL	B
5XSYBA	31	ASP	B
5XSYBA	45	ILE	B
5XSYBA	47	CYS	B
5XSYBA	49	MET	B
5XSYBA	50	ARG	B
5XSYBA	105	GLN	B
5XSYBA	106	ASP	B
5XSYBA	132	PHE	B
5XSYBA	135	TYR	B
5XSYBA	137	PHE	B
5XSYBA	158	ALA	B
5XSYBA	159	SER	B
5XSYBA	162	SER	B
5XSYBA	166	MET	B
5XSYBA	170	ILE	B
5XSYBA	173	LEU	B
5XSYBA	174	GLN	B
5XSYBA	177	LEU	B
5XSYBA	180	GLU	B
5XSYBA	181	MET	B
5XSYBA	184	CYS	B
5XSYBA	185	TYR	B
5XSYBA	189	ALA	B

Scripts

The Chimera scripts are documented here which were applied throughout the study.

```
< Protein - Protein Interface = PPI detection >
measure buriedArea #0.1 #0.2
>Insert formula: sum(@.ASaS)
Select and classif residues
```

```
<Superposition of PPIs by Match Maker >
open proteins 1 and 2
commands:
match #2 to #1 bring #3
swapa (aminoacid name) #1 or #2:Number.(Chain)
```

```
mm #2 to #1/a pair bs alg sw matrix PAM-150 ss false cut 5.0
matrix similarity-matrix
The similarity-matrix can be any of: BLOSUM-30, BLOSUM-35, BLOSUM-40, BLOSUM-45, BLOSUM-50, BLOSUM-55, BLOSUM-60, BLOSUM-62 (default), BLOSUM-65, BLOSUM-70, BLOSUM-75, BLOSUM-80, BLOSUM-85, BLOSUM-90, BLOSUM-100, BLOSUM-N, PAM-40, PAM-120, PAM-150, PAM-250, SDM, HSDM, Nucleic.
```

```
<Generation of PPI surface, buried surface area and calculate the area>
measure buriedArea #0.1 #0.2
>Tools>Structure analysis>Attribute calculator
>Insert formula: sum(atom.areaSES)
Interf
select @/buriedSESArea> 1
Apply
```

Algorithms in phyton

```
Used by methods that color a surface and automatically update the coloring.
# Calling this routine turns off the previous auto-coloring code.
#
def set_coloring_method(name, model, stop_cb = None):

    if hasattr(model, 'coloring_method'):
        cname, cb = model.coloring_method
        if name != cname:
            if cb:
```

```

        cb(model)
    model.coloring_method = (name, stop_cb)

# -----
# Used by methods that adjust surface visibility automatically.
# Calling this routine turns off the previous visibility-adjusting code.
#
def set_visibility_method(name, model, stop_cb = None):

    if hasattr(model, 'visibility_method'):
        cname, cb = model.visibility_method
        if name != cname:
            if cb:
                cb(model)
            model.visibility_method = (name, stop_cb)

# -----
#
def visibility_method(model):

    if hasattr(model, 'visibility_method'):
        cname, cb = model.visibility_method
        return cname
    return None

# -----
#
def visibility_updating(model):

    if hasattr(model, 'visibility_method'):
        cname, cb = model.visibility_method
        return not cb is None
    return False

# -----
#
def reshape_surface(model):

    import chimera
    if isinstance(model, chimera.MSModel):
        model.visibilityMode = model.ByAtom
    else:
        for p in model.surfacePieces:
            show_all_triangles(p)

# -----
#
def show_all_triangles(p):

    mask = p.triangleAndEdgeMask
    if not mask is None:
        mask |= 0x8
        p.triangleAndEdgeMask = mask

# -----
#
# TODO: Make set_triangle_mask() surface piece method.
#
def set_triangle_mask(p, m):

    mask = p.triangleAndEdgeMask
    if mask is None:
        mask = m * 0xf
    else:
        mask &= 0x7
        mask |= (m * 0x8)
    p.triangleAndEdgeMask = mask

# -----
#
def surface_models_with_id(model_id, require_one = False, at_most_one = False):

    from chimera import openModels as om
    if model_id is None or model_id == (om.Default, om.Default):
        return (None if require_one or at_most_one else [])

    from _surface import SurfaceModel
    id, subid = model_id
    slist = om.list(id = id, subid = subid, modelTypes = [SurfaceModel])
    if require_one or at_most_one:
        if len(slist) == 0:
            if at_most_one:
                return None
            mid = '%d' % id if subid == om.Default else '%d.%d' % (id, subid)
            from Commands import CommandError
            raise CommandError('no surfaces with model id %s' % mid)
        elif len(slist) > 1:
            mid = '%d' % id if subid == om.Default else '%d.%d' % (id, subid)
            from Commands import CommandError
            raise CommandError('multiple surfaces with model id %s' % mid)
        return slist[0]

    return slist

# -----
#

```

```

def selection_surface_distance():
    from chimera import selection
    alist = selection.currentAtoms()
    if len(alist) == 0:
        from chimera.replyobj import status
        status('No atoms or markers selected')
        return

    from chimera import openModels
    from _surface import SurfaceModel
    smlist = openModels.list(modelTypes = [SurfaceModel])

    write_surface_distances(alist, smlist)

# -----
#
def write_surface_distances(atoms, surfaces):
    sdisp = [s for s in surfaces if s.display]
    dshown = False
    for s in sdisp:
        xyz = atom_coordinates(atoms, s.openState.xform)
        dist = distance_to_surface(xyz, s)
        for a,d in enumerate(dist):
            print ('Distance from %s to surface %s'
                  % (atoms[a].oslIdent(), s.name))
            print (' d = %.5g, surface point (%.5g, %.5g, %.5g), side %.0f'
                  % tuple(d))
            atoms[a].distance = d[0]
            from chimera.replyobj import status
            status('d = %.5g' % d[0])
            dshown = True
    #     show_line_segments(xyz, dist[1:4])

    from SimpleSession import registerAttribute
    import chimera
    registerAttribute(chimera.Atom, 'distance') # Save distances in session files

    if not dshown:
        from chimera.replyobj import status
        status('No displayed surfaces')
        return

# -----
#
def atom_coordinates(atoms, xform):
    from _multiscale import get_atom_coordinates
    xyz = get_atom_coordinates(atoms, transformed = True)
    from Matrix import xform_matrix
    tf = xform_matrix(xform.inverse())
    from _contour import affine_transform_vertices
    affine_transform_vertices(xyz, tf)
    return xyz

# -----
#
def distance_to_surface(xyz, smodel):
    from _surface import surface_distance
    dist = None
    for p in smodel.surfacePieces:
        if p.display:
            varray, tarray = p.maskedGeometry(p.Solid)
            dist = surface_distance(xyz, varray, tarray, dist)
    return dist

# -----
#
# Compute the closest distance from a point to a surface. The surface is
# represented as a list of triangles. The distance, closest point, and
# side of the closest triangle that the given point lies on is returned.
# Side +1 is the right-handed normal clockwise vertex traversal, while -1
# indicates the opposite side. This is for determining if the given point
# is inside or outside the surface.
#
def surface_distance(xyz, surf_vertex_xyz, surf_triangles, dist = None):
    if dist is None:
        from numpy import zeros, single as floatc
        dist = zeros((len(xyz),5), floatc)
        dist[:,0] = 1e38

    for i,p in enumerate(xyz):
        di = dist[i,:]
        for vertex_indices in surf_triangles:
            t = [surf_vertex_xyz[v] for v in vertex_indices]
            d, txyz, tside = triangle_distance(p, t)
            if d != None and d < di[0]:
                di[0] = d
                di[1:4] = txyz
                di[4] = tside

    return dist

```

```

# -----
# Returns distance, triangle point, and side (1 or -1) of triangle.
#
def triangle_distance(xyz, triangle_vertices):

    # Check if point projects to triangle interior
    va, vb, vc = triangle_vertices
    vab = subtract(vb, va)
    vac = subtract(vb, va)
    vbc = subtract(vb, va)
    n = normalize(cross(vab, vac))
    if n == None:
        return None, None, None          # degenerate triangle

    v = subtract(xyz, va)
    nv = dot(n, v)
    vt = subtract(v, multiply(nv, n))
    if (dot(cross(n, vab), vt) >= 0 and
        dot(cross(vac, n), vt) >= 0 and
        dot(cross(n, vbc), subtract(vt, vab)) >= 0):
        sxyz = add(va, vt)
        d = distance(xyz, sxyz)
        return d, sxyz, sign(nv)

    # Check if point projects onto any edge
    d = None
    for (v1, v2) in ((va, vb), (va, vc), (vb, vc)):
        v12 = subtract(v2, v1)
        v12n2 = dot(v12, v12)
        v = subtract(xyz, v1)
        vv12 = dot(v, v12)
        if vv12 >= 0 and vv12 <= v12n2:
            exyz = add(v1, multiply(vv12/v12n2, v12))
            de = distance(xyz, exyz)
            if d is None or de < d:
                d = de
                sxyz = exyz

    # Check the closest triangle vertex
    vdist = [(distance(xyz, tv), tuple(tv)) for tv in triangle_vertices]
    vdist.sort()
    dv, v = vdist[0]
    if d is None or dv < d:
        d = dv
        sxyz = v

    return d, sxyz, sign(nv)

# -----
#
def distance(u,v):
    w0, w1, w2 = u[0]-v[0], u[1]-v[1], u[2]-v[2]
    import math
    return math.sqrt(w0*w0+w1*w1+w2*w2)

# -----
#
def dot(u,v):
    return u[0]*v[0]+u[1]*v[1]+u[2]*v[2]

# -----
#
def add(u,v):
    return (u[0]+v[0], u[1]+v[1], u[2]+v[2])

# -----
#
def subtract(u,v):
    return (u[0]-v[0], u[1]-v[1], u[2]-v[2])

# -----
#
def multiply(a,v):
    return (a*v[0], a*v[1], a*v[2])

# -----
#
def cross(u,v):
    return (u[1]*v[2]-u[2]*v[1],
            u[2]*v[0]-u[0]*v[2],
            u[0]*v[1]-u[1]*v[0])

# -----
#
def normalize(v):
    import math
    n = math.sqrt(v[0]*v[0]+v[1]*v[1]+v[2]*v[2])
    if n == 0:
        return None
    return (v[0]/n, v[1]/n, v[2]/n)

# -----
#
def sign(x):
    if x >= 0:

```



```

        return 1
    return -1

# -----
#
def show_line_segments(v1, v2, color = (1,1,1,1), surface_model = None):

    n = len(v1)
    from numpy import empty, single as floatc, intc, arange
    varray = empty((2*n,3), floatc)
    tarray = empty((n,3), intc)
    varray[:n,:] = v1
    varray[n:,:] = v2
    tarray[:,0] = tarray[:,2] = arange(n)
    tarray[:,1] = tarray[:,0] + n

    create_model = (surface_model is None)
    if create_model:
        from _surface import SurfaceModel
        surface_model = SurfaceModel()

    p = surface_model.addPiece(varray, tarray, color)
    p.displayStyle = p.Mesh

    if create_model:
        from chimera import openModels as om
        om.add([surface_model])

    return p

# -----
#
class Volume_Manager:

    def __init__(self):

        self.data_regions = []
        self.data_to_regions = {}

        import defaultsettings as d
        self.default_settings = ds = d.Volume_Viewer_Default_Settings()

        # Set default data cache size.
        from VolumeData import data_cache
        data_cache.resize(ds['data_cache_size'] * (2**20))

        self.open_callbacks = []
        self.close_callbacks = []
        self.save_session_callbacks = []

        from chimera import openModels as om
        self.open_handler = om.addAddHandler(self.open_models_cb, None)

        from chimera import triggers, CLOSE_SESSION
        from SimpleSession import SAVE_SESSION
        triggers.addHandler(SAVE_SESSION, self.save_session_cb, None)
        triggers.addHandler(CLOSE_SESSION, self.close_session_cb, None)

        from chimera import SCENE_TOOL_SAVE, SCENE_TOOL_RESTORE
        from session import save_scene, restore_scene
        triggers.addHandler(SCENE_TOOL_SAVE,
                            lambda t,vm,s: save_scene(vm,s), self)
        triggers.addHandler(SCENE_TOOL_RESTORE,
                            lambda t,vm,s: restore_scene(vm,s), self)

# -----
#
def add_volume_opened_callback(self, volume_opened_cb):

    self.open_callbacks.append(volume_opened_cb)

# -----
#
def add_volume_closed_callback(self, volume_closed_cb):

    self.close_callbacks.append(volume_closed_cb)

# -----
#
# Callbacks are called after data has been saved.
# State saved via a callback registered with this routine is restored
# only after volume data has been restored.
#
def add_session_save_callback(self, save_session):

    self.save_session_callbacks.append(save_session)

# -----
#
def open_models_cb(self, trigger_name, args, models):

    vlist = [v for v in models if isinstance(v, Volume)]
    if vlist:
        self.add_volumes(vlist)

# -----

```

```

#
def add_volumes(self, vlist):

    self.data_regions.extend(vlist)

    import chimera
    d2r = self.data_to_regions
    for v in vlist:
        data = v.data
        if data in d2r:
            d2r[data].append(v)
        else:
            d2r[data] = [v]
            chimera.addModelClosedCallback(v, self.model_closed_cb)

    for cb in self.open_callbacks:
        cb(vlist)

# -----
#
def add_volume(self, v):
    self.add_volumes([v])

# -----
#
def set_initial_volume_color(self, v):

    ds = self.default_settings
    if ds['use_initial_colors']:
        n = len(volume_list())
        if v in volume_list():
            n -= 1
        icolors = ds['initial_colors']
        rgba = icolors[n%len(icolors)]
        v.set_parameters(default_rgba = rgba)

# -----
#
def replace_data(self, data, new_data):

    d2r = self.data_to_regions
    d2r[new_data] = d2r[data]
    del d2r[data]
    for dr in d2r[new_data]:
        dr.replace_data(new_data)

# -----
#
def volume_list(self):

    return self.data_regions

# -----
#
def regions_using_data(self, data):

    return self.data_to_regions.get(data, [])

# -----
#
def data_already_opened(self, path, grid_id):

    if not path:
        return None

    dlist = self.data_to_regions.keys()
    for data in dlist:
        if not data.writable and data.path == path and data.grid_id == grid_id:
            return data
    return None

# -----
#
def model_closed_cb(self, v):

    if v in self.data_regions:
        self.remove_volumes([v])

# -----
#
def save_session_cb(self, trigger, x, file):

    import session
    session.save_volume_data_state(self, file)

    for cb in self.save_session_callbacks:
        cb(file)

# -----
#
def close_session_cb(self, trigger, a1, a2):

    self.remove_volumes(self.data_regions)

# -----

```

```

#
def remove_volumes(self, volumes):

    remove_volumes = tuple(volumes)

    for v in remove_volumes:
        index = self.data_regions.index(v)
        del self.data_regions[index]

        data = v.data
        if data is None:
            # Volume has already been closed and data attribute set to None.
            data = [d for d,vlist in self.data_to_regions.items() if v in vlist][0]
        vlist = self.data_to_regions[data]
        vlist.remove(v)
        if len(vlist) == 0:
            del self.data_to_regions[data]
            data.clear_cache()

    for cb in self.close_callbacks:
        cb(remove_volumes)

    # Close models only after removing from volume list.
    # This stops the model closed callback from recalling this routine.
    for v in remove_volumes:
        v.close()

# -----
# Decide whether a data region is small enough to show when opened.
#
def show_when_opened(data_region, show_on_open, max_voxels):

    if not show_on_open:
        return False

    if max_voxels == None:
        return False

    voxel_limit = int(max_voxels * (2 ** 20))
    ss_origin, ss_size, subsampling, ss_step = data_region.ijk_region()
    voxels = float(ss_size[0]) * float(ss_size[1]) * float(ss_size[2])

    return (voxels <= voxel_limit)

# -----
# Decide whether a data region is large enough that only a single z plane
# should be shown.
#
def show_one_plane(size, show_plane, min_voxels):

    if not show_plane:
        return False

    if min_voxels == None:
        return False

    voxel_limit = int(min_voxels * (2 ** 20))
    voxels = float(size[0]) * float(size[1]) * float(size[2])

    return (voxels >= voxel_limit)

# -----
# Manages surface and volume display for a region of a data set.
# Holds surface and solid thresholds, color, and transparency and brightness
# factors.
#
from _surface import SurfaceModel
class Volume(SurfaceModel):

    def __init__(self, data, region = None, rendering_options = None,
                 model_id = None, open_model = True, message_cb = None):

        SurfaceModel.__init__(self)

        from chimera import Material
        self.material = Material.lookup("default")

        self.data = data
        data.add_change_callback(self.data_changed_cb)

        if region is None:
            region = full_region(data.size)
            self.region = clamp_region(region, data.size)

        # C++ Model object raises an error if a unicode name is set.
        # Encode in utf-8 to get a str object that the C++ code can handle.
        self.name = utf8_string(self.data.name)

        if rendering_options is None:
            rendering_options = Rendering_Options()
            self.rendering_options = rendering_options

        self.message_cb = message_cb

```

```

self.matrix_stats = None
self.matrix_id = 1          # Incremented when shape or values change.

rlist = Region_List()
ijk_min, ijk_max = self.region[:2]
rlist.insert_region(ijk_min, ijk_max)
self.region_list = rlist

self.representation = 'surface'

self.solid = None
self.keep_displayed_data = None

self.display = False      # Undisplay surface

self.initialized_thresholds = False

# Surface display parameters
self.surface_levels = []
self.surface_colors = []
self.surface_piece_list = [] # SurfacePiece graphics objects
self.surface_brightness_factor = 1
self.transparency_factor = 0 # for surface/mesh
self.outline_box = Outline_Box(self)

# Solid display parameters
self.solid_levels = [] # list of (threshold, scale)
self.solid_colors = []
self.transparency_depth = 0.5 # for solid
self.solid_brightness_factor = 1

self.default_rgba = data.rgba if data.rgba else (.7,.7,.7,1)

self.change_callbacks = []

if open_model:
    self.open_model(model_id)

from chimera import addModelClosedCallback
addModelClosedCallback(self, self.model_closed_cb)

from chimera import triggers
h = triggers.addHandler('SurfacePiece', self.surface_piece_changed_cb, None)
self.surface_piece_change_handler = h

# -----
#
def message(self, text):

    if self.message_cb:
        self.message_cb(text)

# -----
#
def full_name(self):

    return self.name

# -----
#
def name_with_id(self):

    if self.subid:
        sid = '#%d.%d' % (self.id, self.subid)
    else:
        sid = '#%d' % self.id
    return self.name + '_' + sid

# -----
#
def add_volume_change_callback(self, cb):

    self.change_callbacks.append(cb)

# -----
#
def remove_volume_change_callback(self, cb):

    self.change_callbacks.remove(cb)

# -----
#
def call_change_callbacks(self, change_types):

    if isinstance(change_types, basestring):
        change_types = [change_types]

    for cb in self.change_callbacks:
        for ct in change_types:
            cb(self, ct)

# -----
# Detect surface piece color change and display style change.
#
def surface_piece_changed_cb(self, trigger, unused, changes):

```

```

change = ('color changed' in changes.reasons
         or 'display style changed' in changes.reasons)
if not change:
    return

if len(self.surface_piece_list) != len(self.surface_colors):
    return

pindex = {}
for i, p in enumerate(self.surface_piece_list):
    pindex[p] = i

# Check if surface pieces for this volume have changed.
ctypes = set()
pchange = False
for p in changes.modified:
    if p in pindex:
        i = pindex[p]
        vcolor = self.modulated_surface_color(self.surface_colors[i])
        from numpy import array, single as floatc
        if (p.color != array(vcolor, floatc)).any():
            self.surface_colors[i] = p.color
            ctypes.add('colors changed')
        pchange = True

# Check if display style of all surface pieces has changed.
if pchange and self.representation != 'solid':
    styles = set()
    for p in self.surface_piece_list:
        styles.add(p.displayStyle)
    if len(styles) == 1:
        pstyle = (p.Solid: 'surface', p.Mesh: 'mesh').get(styles.pop(), None)
        if pstyle and self.representation != pstyle:
            # TODO: Eliminate special case for 2d contour rendering.
            contour_2d = (pstyle == 'mesh' and
                          not self.rendering_options.cap_faces
                          and self.single_plane())
            if not contour_2d:
                self.set_representation(pstyle)

# Notify about changes.
self.call_change_callbacks(ctypes)

# -----
#
def replace_data(self, data):

    d = self.data
    cc = (data.origin != d.origin or
          data.step != d.step or
          data.cell_angles != d.cell_angles)
    dc = self.data_changed_cb
    d.remove_change_callback(dc)
    self.data = data
    data.add_change_callback(dc)
    dc('values changed')
    if cc:
        dc('coordinates changed')

# -----
# Programatically set data region parameters. The following keyword
# parameters are valid.
#
# surface_levels
# surface_colors          (rgb or rgba values)
# surface_brightness_factor
# transparency_factor
# solid_levels
# solid_colors           (rgb or rgba values)
# transparency_depth
# solid_brightness_factor
#
# Any rendering option attribute names can also be used.
#
# The volume display is not automatically updated. Use v.show() when
# redisplay is desired.
#
def set_parameters(self, **kw):

    parameters = ('surface_levels',
                  'surface_colors',
                  'surface_brightness_factor',
                  'transparency_factor',
                  'solid_levels',
                  'solid_colors',
                  'solid_brightness_factor',
                  'transparency_depth',
                  'default_rgba',
                  )

def rgb_to_rgba(color):
    if len(color) == 3:
        return tuple(color) + (1,)
    return color

```

```

if 'surface_colors' in kw:
    kw['surface_colors'] = map(rgb_to_rgba, kw['surface_colors'])
if 'solid_colors' in kw:
    kw['solid_colors'] = map(rgb_to_rgba, kw['solid_colors'])

if ('surface_levels' in kw and
    not 'surface_colors' in kw and
    len(kw['surface_levels']) != len(self.surface_levels)):
    kw['surface_colors'] = [self.default_rgba] * len(kw['surface_levels'])
if ('solid_levels' in kw and
    not 'solid_colors' in kw and
    len(kw['solid_levels']) != len(self.solid_colors)):
    rgba = saturate_rgba(self.default_rgba)
    kw['solid_colors'] = [rgba] * len(kw['solid_levels'])

if 'default_rgba' in kw:
    self.default_rgba = kw['default_rgba'] = rgb_to_rgba(kw['default_rgba'])

# Make copies of lists.
for param in ('surface_levels', 'surface_colors',
              'solid_levels', 'solid_colors'):
    if param in kw:
        kw[param] = list(kw[param])

for param in parameters:
    if param in kw:
        values = kw[param]
        setattr(self, param, values)

# Update rendering options.
option_changed = False
if ('orthoplanes_shown' in kw and not 'box_faces' in kw and
    true_count(kw['orthoplanes_shown']) > 0):
    # Turn off box faces if orthoplanes enabled.
    kw['box_faces'] = False
ro = self.rendering_options
box_faces_toggled = ('box_faces' in kw and kw['box_faces'] != ro.box_faces)
orthoplanes_toggled = ('orthoplanes_shown' in kw and
    kw['orthoplanes_shown'] != ro.orthoplanes_shown and
    (true_count(kw['orthoplanes_shown']) == 0 or
     true_count(ro.orthoplanes_shown) == 0))
adjust_step = (self.representation == 'solid' and
    (box_faces_toggled or orthoplanes_toggled))
for k,v in kw.items():
    if k in ro.__dict__:
        setattr(self.rendering_options, k, v)
        option_changed = True
if adjust_step:
    r = self.region
    self.new_region(r[0], r[1], r[2], show = False, adjust_step = True,
        save_in_region_queue = False)

if 'surface_levels' in kw or 'solid_levels' in kw:
    self.call_change_callbacks('thresholds changed')

if ('surface_colors' in kw or 'solid_colors' in kw or
    'surface_brightness_factor' in kw or 'transparency_factor' in kw or
    'solid_brightness_factor' in kw or 'transparency_depth' in kw):
    self.call_change_callbacks('colors changed')

if option_changed:
    self.call_change_callbacks('rendering options changed')

# -----
# Sets new region and optionally shows it.
#
def new_region(self, ijk_min = None, ijk_max = None, ijk_step = None,
    show = True, adjust_step = True, save_in_region_queue = True):

    if ijk_min is None:
        ijk_min = self.region[0]
    if ijk_max is None:
        ijk_max = self.region[1]

    # Make bounds integers.
    from math import ceil, floor
    ijk_min = [int(ceil(x)) for x in ijk_min]
    ijk_max = [int(floor(x)) for x in ijk_max]

    # Make it lie within dota bounds.
    (ijk_min, ijk_max) = clamp_region((ijk_min, ijk_max), self.data.size)

    # Determine ijk_step.
    if ijk_step == None:
        if self.region:
            ijk_step = self.region[2]
        else:
            ijk_step = (1,1,1)
    else:
        ijk_step = [int(ceil(x)) for x in ijk_step]

    # Adjust ijk_step to meet voxel limit.
    ro = self.rendering_options
    fpa = faces_per_axis(self.representation, ro.box_faces,

```

```

        ro.any_orthoplanes_shown())
adjusted_ijk_step = ijk_step_for_voxel_limit(ijk_min, ijk_max, ijk_step,
                                             fpa, ro.limit_voxel_count,
                                             ro.voxel_limit)

if adjust_step:
    ijk_step = adjusted_ijk_step
elif tuple(ijk_step) != tuple(adjusted_ijk_step):
    # Change automatic step adjustment voxel limit.
    vc = subarray_size(ijk_min, ijk_max, ijk_step, fpa)
    ro.voxel_limit = (1.01*vc) / (2**20) # Mvoxels rounded up for gui value
    self.call_change_callbacks('voxel limit changed')

if save_in_region_queue:
    self.region_list.insert_region(ijk_min, ijk_max)

region = (ijk_min, ijk_max, ijk_step)
if same_region(region, self.region):
    return False

self.region = region
self.matrix_changed()

self.call_change_callbacks('region changed')

if show:
    self.show()

return True

# -----
#
def is_full_region(self, region = None):

    if region is None:
        region = self.region
    elif region == 'all':
        return True
    ijk_min, ijk_max, ijk_step = region
    dmax = tuple([s-1 for s in self.data.size])
    full = (tuple(ijk_min) == (0,0,0) and
            tuple(ijk_max) == dmax and
            tuple(ijk_step) == (1,1,1))
    return full

# -----
# Either data values or subregion has changed.
#
def matrix_changed(self):

    self.matrix_stats = None
    self.matrix_id += 1

# -----
#
def has_thresholds(self):

    return len(self.surface_levels) > 0 and len(self.solid_levels) > 0

# -----
Matchmaker scripts & Algorithms
$Id: __init__.py 39819 2014-05-20 00:16:46Z pett $

import SmithWaterman
import NeedlemanWunsch
from chimera.Sequence import StructureSequence
from chimera import replyobj, UserError, LimitationError

CP_SPECIFIC_SPECIFIC = "ss"
CP_SPECIFIC_BEST = "sb"
CP_BEST = "bb"

SA_NEEDLEMAN_WUNSCH = "Needleman-Wunsch"
SA_SMITH_WATERMAN = "Smith-Waterman"

from prefs import defaults, CHAIN_PAIRING, SEQUENCE_ALGORITHM, MATRIX, \
    GAP_OPEN, GAP_EXTEND, SS_MIXTURE, SS_SCORES, ITER_CUTOFF, HELIX_OPEN, \
    STRAND_OPEN, OTHER_OPEN, COMPUTE_SS
defaultSSMatrix = defaults[SS_SCORES]

# called recursively, so any changes to calling signature need to happen
# in recursive call too...
def align(ref, match, matrix, algorithm, gapOpen, gapExtend, ksdsspCache,
          ssMatrix=defaults[SS_SCORES],
          ssFraction=defaults[SS_MIXTURE],
          gapOpenHelix=defaults[HELIX_OPEN],
          gapOpenStrand=defaults[STRAND_OPEN],
          gapOpenOther=defaults[OTHER_OPEN],
          computeSS=defaults[COMPUTE_SS]):

    similarityMatrix = SmithWaterman.matrices[matrix]
    ssf = ssFraction
    ssm = ssMatrix
    if ssf is not None and ssf is not False and computeSS:
        needCompute = []

```



```

if ref.molecule not in ksdsspCache:
    for r in ref.residues:
        if r and len(r.atoms) > 1:
            # not CA only
            needCompute.append(ref.molecule)
            ksdsspCache.add(ref.molecule)
            break
if match.molecule not in ksdsspCache:
    for r in match.residues:
        if r and len(r.atoms) > 1:
            # not CA only
            needCompute.append(match.molecule)
            ksdsspCache.add(match.molecule)
            break
if needCompute:
    from chimera.initprefs import ksdsspPrefs, \
        KSDSSP_ENERGY, KSDSSP_HELIX_LENGTH, \
        KSDSSP_STRAND_LENGTH
    from Midas import ksdssp
    ksdssp(needCompute, energy=ksdsspPrefs[KSDSSP_ENERGY],
           helixLen=ksdsspPrefs[KSDSSP_HELIX_LENGTH],
           strandLen=ksdsspPrefs[KSDSSP_STRAND_LENGTH])
if algorithm == "nw":
    score, seqs = NeedlemanWunsch.nw(ref, match,
                                     scoreGap=-gapExtend, scoreGapOpen=0-gapOpen,
                                     similarityMatrix=similarityMatrix, returnSeqs=True,
                                     ssMatrix=ssMatrix, ssFraction=ssFraction,
                                     gapOpenHelix=-gapOpenHelix,
                                     gapOpenStrand=-gapOpenStrand,
                                     gapOpenOther=-gapOpenOther)
    gappedRef, gappedMatch = seqs
elif algorithm == "sw":
    refName = ref.molecule.name
    if not ref.name.startswith("principal"):
        refName += ", " + ref.name
    gappedRef = StructureSequence(ref.molecule, refName)
    matchName = match.molecule.name
    if not match.name.startswith("principal"):
        matchName += ", " + match.name
    gappedMatch = StructureSequence(match.molecule, matchName)
    def ssLet(r):
        if not r:
            return ' '
        if r.isHelix:
            return 'H'
        elif r.isStrand:
            return 'S'
        return 'O'
    if ssf is False or ssf is None:
        ssf = 0.0
        ssm = None
    if ssm:
        # account for missing structure (blank SS letter)
        ssm = ssm.copy()
        for let in "HSO ":
            ssm[(let, ' ')] = 0.0
            ssm[( ' ', let)] = 0.0
    score, alignment = SmithWaterman.align(str(ref), str(match),
                                           similarityMatrix, float(gapOpen), float(gapExtend),
                                           gapChar=".", ssMatrix=ssm, ssFraction=ssf,
                                           gapOpenHelix=float(gapOpenHelix),
                                           gapOpenStrand=float(gapOpenStrand),
                                           gapOpenOther=float(gapOpenOther),
                                           ss1="".join([ssLet(r) for r in ref.residues]),
                                           ss2="".join([ssLet(r) for r in match.residues]))
    gappedRef.extend(alignment[0])
    gappedMatch.extend(alignment[1])
    # Smith-Waterman may not be entirety of sequences...
    for orig, gapped in [(ref, gappedRef), (match, gappedMatch)]:
        ungapped = gapped.ungapped()
        for i in range(len(orig) - len(ungapped) + 1):
            if ungapped == orig[i:i+len(ungapped)]:
                break
        else:
            raise ValueError("Smith-Waterman result not"
                             " a subsequence of original sequence")
        gapped.residues = orig.residues[i:i+len(ungapped)]
        resMap = {}
        gapped.resMap = resMap
        for j in range(len(ungapped)):
            gres = gapped.residues[j]
            if gres:
                resMap[gres] = j
    else:
        raise ValueError("Unknown sequence alignment algorithm: %s"
                         % algorithm)

# If the structures are disjoint snippets of the same longer SEQRES,
# they may be able to be structurally aligned but the SEQRES records
# will keep them apart. Try to detect this situation and work around
# by snipping off sequence ends.
srDisjoint = False
if 'SEQRES' in ref.molecule.pdbHeaders and 'SEQRES' in match.molecule.pdbHeaders:
    structMatch = 0
    for i in range(len(gappedRef)):

```

```

        uri = gappedRef.gapped2ungapped(i)
        if uri is None:
            continue
        umi = gappedMatch.gapped2ungapped(i)
        if umi is None:
            continue
        if gappedRef.residues[uri] and gappedMatch.residues[umi]:
            structMatch += 1
            if structMatch >= 3:
                break
    if structMatch < 3:
        seqMatch = 0
        for s1, s2 in zip(gappedRef[:], gappedMatch[:]):
            if s1.isalpha() and s2.isalpha():
                seqMatch += 1
                if seqMatch > 3:
                    break
    if seqMatch > 3:
        need = 3 - structMatch
        if (ref.residues[:need].count(None) == 3
            or ref.residues[-need:].count(None) == 3) \
            and (match.residues[:need].count(None) == 3
                or match.residues[-need:].count(None) == 3):
            srDisjoint = True
if srDisjoint:
    from copy import copy
    clippedRef = copy(ref)
    clippedMatch = copy(match)
    for seq in (clippedRef, clippedMatch):
        numNone = 0
        for r in seq.residues:
            if r:
                break
            numNone += 1
        if numNone:
            seq[:] = seq[numNone:]
            seq.residues = seq.residues[numNone:]
            for r, i in seq.resMap.items():
                seq.resMap[r] = i - numNone
        numNone = 0
        for r in reversed(seq.residues):
            if r:
                break
            numNone += 1
        if numNone:
            seq[:] = seq[:-numNone]
            seq.residues = seq.residues[:-numNone]
    return align(clippedRef, clippedMatch, matrix, algorithm, gapOpen,
                 gapExtend, ksdsspCache, ssMatrix=ssMatrix, ssFraction=ssFraction,
                 gapOpenHelix=gapOpenHelix, gapOpenStrand=gapOpenStrand,
                 gapOpenOther=gapOpenOther, computeSS=False)
for orig, aligned in [(ref, gappedRef), (match, gappedMatch)]:
    if hasattr(orig, '_dmRebuildInfo'):
        aligned._dmRebuildInfo = orig._dmRebuildInfo
        _dmCleanup.append(aligned)
return score, gappedRef, gappedMatch

def matrixCompatible(chain, matrix):
    proteinMatrix = len(SmithWaterman.matrices[matrix]) >= 400
    return proteinMatrix == chain.hasProtein()

def match(chainPairing, matchItems, matrix, alg, gapOpen, gapExtend, iterate=None,
          showAlignment=False, align=align, domainResidues=(None, None),
          verbose=False, **alignKw):
    """Superimpose structures based on sequence alignment

    'chainPairing' is the method of pairing chains to match:

    CP_SPECIFIC_SPECIFIC --
    Each reference chain is paired with a specified match chain

    CP_SPECIFIC_BEST --
    Single reference chain is paired with best seq-aligning
    chain from one or more molecules

    CP_BEST --
    Best seq-aligning pair of chains from reference molecule and
    match molecule(s) is used
    """
    ksdsspCache = set()
    alg = alg.lower()
    if alg == "nw" or alg.startswith("needle"):
        alg = "nw"
        algName = "Needleman-Wunsch"
    elif alg == "sw" or alg.startswith("smith"):
        alg = "sw"
        algName = "Smith-Waterman"
    else:
        raise ValueError("Unknown sequence alignment algorithm: %s"
                         % alg)

    pairings = {}
    smallMolErrMsg = "Reference and/or match model contains no nucleic or" \
        " amino acid chains.\nUse the command-line 'match' command" \
        " to superimpose small molecules/ligands."

```

```

rdRes, mdRes = domainResidues
if chainPairing == CP_SPECIFIC_SPECIFIC:
    # specific chain(s) in each

    # various sanity checks
    #
    # (1) can't have same chain matched to multiple refs
    # (2) reference molecule can't be a match molecule
    matchChains = {}
    matchMols = {}
    refMols = {}
    for ref, match in matchItems:
        if not matrixCompatible(ref, matrix):
            raise UserError("Reference chain (%s) not"
                             " compatible with %s similarity"
                             " matrix" % (ref.fullName(), matrix))
        if not matrixCompatible(match, matrix):
            raise UserError("Match chain (%s) not"
                             " compatible with %s similarity"
                             " matrix" % (match.fullName(), matrix))
        if match in matchChains:
            raise UserError("Cannot match the same chain"
                             " to multiple reference chains")
        matchChains[match] = ref
        if match.molecule in refMols \
        or ref.molecule in matchMols \
        or match.molecule == ref.molecule:
            raise UserError("Cannot have same molecule"
                             " model provide both reference and"
                             " match chains")
        matchMols[match.molecule] = ref
        refMols[ref.molecule] = match

    if not matchChains:
        raise UserError("Must select at least one reference"
                         " chain.\n")

    for match, ref in matchChains.items():
        match, ref = [checkDomainMatching([ch], dr)[0] for ch, dr in
                      ((match, mdRes), (ref, rdRes))]
        score, s1, s2 = align(ref, match, matrix, alg,
                              gapOpen, gapExtend,
                              ksdsspCache, **alignKw)
        pairings.setdefault(s2.molecule, []).append(
            (score, s1, s2))

elif chainPairing == CP_SPECIFIC_BEST:
    # specific chain in reference;
    # best seq-aligning chain in match model(s)
    ref, matches = matchItems
    if not ref or not matches:
        raise UserError("Must select at least one reference"
                         " and match item.\n")
    if not matrixCompatible(ref, matrix):
        raise UserError("Reference chain (%s) not compatible"
                         " with %s similarity matrix"
                         " % (ref.fullName(), matrix))
    ref = checkDomainMatching([ref], rdRes)[0]
    for match in matches:
        bestScore = None
        seqs = [s for s in match.sequences()
                if matrixCompatible(s, matrix)]
        if not seqs and match.sequences():
            raise UserError("No chains in match structure"
                             " %s compatible with %s similarity"
                             " matrix" % (match, matrix))
        seqs = checkDomainMatching(seqs, mdRes)
        for seq in seqs:
            score, s1, s2 = align(ref, seq, matrix, alg,
                                  gapOpen, gapExtend,
                                  ksdsspCache, **alignKw)
            if bestScore is None or score > bestScore:
                bestScore = score
                pairing = (score, s1, s2)
        if bestScore is None:
            raise LimitationError(smallMolErrMsg)
        pairings[match] = [pairing]

elif chainPairing == CP_BEST:
    # best seq-aligning pair of chains between
    # reference and match structure(s)
    ref, matches = matchItems
    if not ref or not matches:
        raise UserError("Must select at least one reference"
                         " and match item in different models.\n")
    rseqs = [s for s in checkDomainMatching(ref.sequences(), rdRes)
             if matrixCompatible(s, matrix)]
    if not rseqs and ref.sequences():
        raise UserError("No chains in reference structure"
                         " %s compatible with %s similarity"
                         " matrix" % (ref, matrix))
    for match in matches:
        bestScore = None
        mseqs = [s for s in checkDomainMatching(match.sequences(), mdRes)
                 if matrixCompatible(s, matrix)]

```

```

if not mseqs and match.sequences():
    raise UserError("No chains in match structure"
                    " %s compatible with %s similarity"
                    " matrix" % (match, matrix))
for mseq in mseqs:
    for rseq in rseqs:
        score, s1, s2 = align(rseq, mseq,
                               matrix, alg, gapOpen, gapExtend,
                               ksdsspCache, **alignKw)
        if bestScore is None \
        or score > bestScore:
            bestScore = score
            pairing = (score,s1,s2)
    if bestScore is None:
        raise LimitationError(smallMolErrMsg)
    pairings[match]= [pairing]
else:
    raise ValueError("No such chain-pairing method")

from chimera.misc import principalAtom
retVals = []
for matchMol, pairs in pairings.items():
    refAtoms = []
    matchAtoms = []
    regionInfo = {}
    if verbose:
        seqPairings = []
    for score, s1, s2 in pairs:
        try:
            ssMatrix = alignKw['ssMatrix']
        except KeyError:
            ssMatrix = defaultSSMatrix
        try:
            ssFraction = alignKw['ssFraction']
        except KeyError:
            ssFraction = defaults[SS_MIXTURE]

        replyobj.info("\n")
        replyobj.status("Matchmaker %s (%s) with %s (%s),"
                        " sequence alignment score = %g" % (
                            s1.name, s1.molecule.oslIdent(), s2.name,
                            s2.molecule.oslIdent(), score), log=1)
        replyobj.info("with these parameters:\n"
                      "\tchain pairing: %s\n\t%s using %s\n"
                      % (chainPairing, algName, matrix))

        if ssFraction is None or ssFraction is False:
            replyobj.info("\tno secondary structure"
                          " guidance used\n")
            replyobj.info("\tgap open %g, extend %g\n" % (
                gapOpen, gapExtend))
        else:
            if 'gapOpenHelix' in alignKw:
                gh = alignKw['gapOpenHelix']
            else:
                gh = defaults[HELIX_OPEN]
            if 'gapOpenStrand' in alignKw:
                gs = alignKw['gapOpenStrand']
            else:
                gs = defaults[STRAND_OPEN]
            if 'gapOpenOther' in alignKw:
                go = alignKw['gapOpenOther']
            else:
                go = defaults[OTHER_OPEN]
            replyobj.info("\tss fraction: %g\n"
                          "\tgap open (HH/SS/other) %g/%g/%g, "
                          "extend %g\n"
                          "\tss matrix: " % (ssFraction, gh, gs,
                          go, gapExtend))
            for ss1, ss2 in ssMatrix.keys():
                if ss2 < ss1:
                    continue
                replyobj.info(" (%s, %s): %g" % (ss1,
                    ss2, ssMatrix[(ss1, ss2)]))
            replyobj.info("\n")
        if iterate is None:
            replyobj.info("\tno iteration\n")
        else:
            replyobj.info("\titeration cutoff: %g\n"
                          % iterate)

    skip = set()
    if showAlignment:
        from MultAlignViewer.MAViewer import MAViewer
        for s in [s1,s2]:
            if hasattr(s, '_dmRebuildInfo'):
                for i, c, r in s._dmRebuildInfo:
                    g = s.ungapped2gapped(i)
                    s[g] = c
                    s.residues[i] = r
                    skip.add(r)
                s.resMap.clear()
                for i, r in enumerate(s.residues):
                    if r:
                        s.resMap[r] = i
        mav = MAViewer([s1,s2], autoAssociate=None)

```

```

        mav.autoAssociate = True
        mav.hideHeaders(mav.headers(shownOnly=True))
        from MAVHeader.ChimeraExtension import CaDistanceSeq
        mav.showHeaders([h for h in mav.headers()
                        if h.name == CaDistanceSeq.name])
    for i in range(len(s1)):
        if s1[i] == "." or s2[i] == ".":
            continue
        refRes = s1.residues[s1.gapped2ungapped(i)]
        matchRes = s2.residues[s2.gapped2ungapped(i)]
        if not refRes:
            continue
        refAtom = principalAtom(refRes)
        if not refAtom:
            continue
        if not matchRes:
            continue
        matchAtom = principalAtom(matchRes)
        if not matchAtom:
            continue
        if refRes in skip or matchRes in skip:
            continue
        if refAtom.name != matchAtom.name:
            # nucleic P-only trace vs. full nucleic
            if refAtom.name != "P":
                try:
                    refAtom = refAtom.residue.atomsMap["P"][0]
                except KeyError:
                    continue
        else:
            try:
                matchAtom = matchAtom.residue.atomsMap["P"][0]
            except KeyError:
                continue
        refAtoms.append(refAtom)
        matchAtoms.append(matchAtom)
        if showAlignment and iterate is not None:
            regionInfo[refAtom] = (mav, i)

    if verbose:
        seqPairings.append((s1, s2))
import Midas
if len(matchAtoms) < 3:
    replyobj.error("Fewer than 3 residues aligned; cannot"
                  " match %s with %s\n" % (s1.name, s2.name))
    continue
try:
    retVals.append(Midas.match(matchAtoms, refAtoms,
                              iterate=iterate, minPoints=3))
except Midas.TooFewAtomsError:
    replyobj.error("Iteration produces fewer than 3"
                  " residues aligned.\nCannot match %s with %s"
                  " satisfying iteration threshold.\n"
                  % (s1.name, s2.name))
    continue
replyobj.info("\n") # separate matches with whitespace
if regionInfo:
    byMav = {}
    for ra in retVals[-1][1]:
        mav, index = regionInfo[ra]
        byMav.setdefault(mav, []).append(index)
    for mav, indices in byMav.items():
        indices.sort()
        from MultAlignViewer.MAViewer import \
            MATCHED_REGION_INFO
        name, fill, outline = MATCHED_REGION_INFO
        mav.newRegion(name=name, columns=indices,
                     fill=fill, outline=outline)
        mav.status("Residues used in final fit"
                  " iteration are highlighted")
if verbose:
    for s1, s2 in seqPairings:
        replyobj.info("Sequences:\n")
        for s in [s1, s2]:
            replyobj.info(s.name + "\t" + str(s) + "\n")
        replyobj.info("Residues:\n")
        for s in [s1, s2]:
            replyobj.info(" ".join([str(r) for r in s.residues]) + "\n")
        replyobj.info("Residue usage in match (1=used, 0=unused):\n")
        matchAtoms1, matchAtoms2 = retVals[-1][:2]
        matchResidues = set([a.residue
                            for matched in retVals[-1][:2] for a in matched])
        for s in [s1, s2]:
            replyobj.info(" ".join([str(int(r in matchResidues))
                                    for r in s.residues]) + "\n")

global _dmCleanup
for seq in _dmCleanup:
    delattr(seq, '_dmRebuildInfo')
_dmCleanup = []
return retVals

def cmdMatch(refSel, matchSel, pairing=defaults[CHAIN_PAIRING],
             alg=defaults[SEQUENCE_ALGORITHM], verbose=False,
             ssFraction=defaults[SS_MIXTURE], matrix=defaults[MATRIX],

```

```

        gapOpen=defaults[GAP_OPEN], hgap=defaults[HELIX_OPEN],
        sgap=defaults[STRAND_OPEN], ogap=defaults[OTHER_OPEN],
        iterate=defaults[ITER_CUTOFF], gapExtend=defaults[GAP_EXTEND],
        showAlignment=False, computeSS=defaults[COMPUTE_SS],
        matHH=defaultSSMatrix[('H', 'H')],
        matSS=defaultSSMatrix[('S', 'S')],
        matOO=defaultSSMatrix[('O', 'O')],
        matHS=defaultSSMatrix[('H', 'S')],
        matHO=defaultSSMatrix[('H', 'O')],
        matSO=defaultSSMatrix[('S', 'O')]):
    """wrapper for command-line command (friendlier args)"""
    from Midas import MidasError
    if matrix not in SmithWaterman.matrices:
        raise MidasError("No such matrix name: %s" % str(matrix))
    try:
        gapOpen + 1
        gapExtend + 1
        hgap + 1
        sgap + 1
        ogap + 1
    except TypeError:
        raise MidasError("Gap open/extend penalties must be numeric")
    if pairing == CP_SPECIFIC_SPECIFIC:
        matches = matchSel.chains(ordered=True)
    elif pairing == CP_SPECIFIC_BEST:
        matches = matchSel.molecules()
    if pairing == CP_SPECIFIC_SPECIFIC or pairing == CP_SPECIFIC_BEST:
        refs = refSel.chains(ordered=True)
        if not refs:
            raise MidasError("No reference chains specified")
        if pairing == CP_SPECIFIC_BEST:
            if len(refs) > 1:
                raise MidasError("Specify a single reference chain only")
    else:
        refMols = refSel.molecules()
        if not refMols:
            raise MidasError("No reference model specified")
        if len(refMols) > 1:
            raise MidasError("Specify a single reference"
                               " model only")
        refs = refMols
        matches = matchSel.molecules()
    if not matches:
        raise MidasError("No molecules/chains to match specified")
    for ref in refs:
        if ref in matches:
            matches.remove(ref)
    if not matches:
        raise MidasError("Must use different reference and match"
                               " structures")
    if pairing == CP_SPECIFIC_SPECIFIC:
        if len(refs) != len(matches):
            raise MidasError("Different number of reference/match"
                               " chains (%d ref, %d match)" %
                               (len(refs), len(matches)))
        matchItems = zip(refs, matches)
    else:
        matchItems = (refs[0], matches)
    ssMatrix = {}
    ssMatrix[('H', 'H')] = float(matHH)
    ssMatrix[('S', 'S')] = float(matSS)
    ssMatrix[('O', 'O')] = float(matOO)
    ssMatrix[('H', 'S')] = ssMatrix[('S', 'H')] = float(matHS)
    ssMatrix[('H', 'O')] = ssMatrix[('O', 'H')] = float(matHO)
    ssMatrix[('S', 'O')] = ssMatrix[('O', 'S')] = float(matSO)
    if type(iterate) == bool and not iterate:
        iterate = None
    try:
        match(pairing, matchItems, matrix, alg, gapOpen, gapExtend,
              ssFraction=ssFraction, ssMatrix=ssMatrix,
              iterate=iterate, showAlignment=showAlignment,
              domainResidues=(refSel.residues(), matchSel.residues()),
              gapOpenHelix=hgap, gapOpenStrand=sgap,
              gapOpenOther=ogap, computeSS=computeSS, verbose=verbose)
    except UserError, v:
        raise MidasError, v

_dmCleanup = []
def checkDomainMatching(chains, selResidues):
    if not selResidues:
        return chains
    chainResidues = set([r for ch in chains for r in ch.residues if r])
    selResidues = set(selResidues)
    if not chainResidues.issubset(selResidues):
        # domain matching
        newChains = []
        for chain in chains:
            thisChain = set([r for r in chain.residues if r])
            if thisChain.issubset(selResidues):
                newChains.append(chain)
            continue
        nc = StructureSequence(chain.molecule, chain.name)
        nc._dmRebuildInfo = []
        _dmCleanup.append(nc)
        newChains.append(nc)

```

```
for c, r in zip(str(chain), chain.residues):
    if r in selResidues:
        nc.append(c)
        nc.resMap[r] = len(nc.residues)
        nc.residues.append(r)
    else:
        nc._dmRebuildInfo.append((len(nc.residues), c, r))
        nc.append('?')
        nc.residues.append(None)
    chains = newChains
return chains
```