



Kandidatutkielma

Tietojenkäsittelytieteen kandiohjelma

# Koneoppiminen lääkkeiden kehityksessä

Heikki Pulli

30.11.2021

MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA  
HELSINGIN YLIOPISTO

**Ohjaaja(t)**

Prof. D.U. Mind, Dr. O. Why

**Tarkastaja(t)**

Prof. D.U. Mind, Dr. O. Why

**Yhteystiedot**

PL 68 (Pietari Kalmin katu 5)  
00014 Helsingin yliopisto

Sähköpostiosoite: [info@cs.helsinki.fi](mailto:info@cs.helsinki.fi)

URL: <http://www.cs.helsinki.fi/>

HELSINGIN YLIOPISTO – HELSINGFORS UNIVERSITET – UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Matemaattis-luonnontieteellinen tiedekunta		Tietojenkäsittelytieteen kandiohjelma	
Tekijä — Författare — Author			
Heikki Pulli			
Työn nimi — Arbetets titel — Title			
Koneoppiminen lääkkeiden kehityksessä			
Ohjaajat — Handledare — Supervisors			
Prof. D.U. Mind, Dr. O. Why			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Kandidutkielma	30.11.2021	20 sivua	
Tiivistelmä — Referat — Abstract			
<p>Tämä dokumentti on tarkoitettu Helsingin yliopiston tietojenkäsittelytieteen osaston opinnäytteiden ja harjoitustöiden ulkoasun ohjeeksi ja mallipohjaksi. Ohje soveltuu kandidutkielmiin, ohjelmistotuotantoprojekteihin, seminaareihin ja maisterintutkielmiin. Tämän ohjeen lisäksi on seurattava niitä ohjeita, jotka opastavat valitsemaan kuhunkin osioon tieteellisesti kiinnostavaa, syvällisesti pohdittua sisältöä.</p> <p>Työn aihe luokitellaan ACM Computing Classification System (CCS) mukaisesti, ks. <a href="https://www.acm.org/about-acm/class">https://www.acm.org/about-acm/class</a>, käyttäen komentoa <code>\classification{}</code>. Käytä muutamaa termipolkua (1–3), jotka alkavat juuritermistä ja joissa polun tarkentuvat luokat erotetaan toisistaan oikealle osoittavalla nuolella.</p> <p><b>ACM Computing Classification System (CCS)</b>    General and reference → Document types → Surveys and overviews  Applied computing → Document management and text processing → Document management → Text editing</p>			
Avainsanat — Nyckelord — Keywords			
ulkoasu, tiivistelmä, lähdeluettelo			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsingin yliopiston kirjasto			
Muita tietoja — övriga uppgifter — Additional information			
Ohjelmistojärjestelmien erikoistumislinja			



# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Kemiallinen data</b>	<b>3</b>
2.1	SMILES, SMARTS ja SMIRKS . . . . .	3
2.2	Ominaisuusvektorit . . . . .	3
<b>3</b>	<b>Uusien lääkkeiden löytäminen</b>	<b>5</b>
3.1	Lääkkeiden löytäminen . . . . .	5
3.2	Virtuaalinen seulonta . . . . .	5
3.3	Kehitettyjä koneoppimismalleja . . . . .	6
3.3.1	Controlled molecule generator, CMG . . . . .	6
3.3.2	Prototyyppiin perustuva lääkekehitys . . . . .	8
3.3.3	CDN-VAE . . . . .	8
<b>4</b>	<b>Uusien lääkkeiden syntetisointi</b>	<b>9</b>
4.1	Yhdisteiden syntetisointi . . . . .	9
4.2	Lääkkeen retrosyntetisoinnin haastavuus . . . . .	10
4.3	Kehitettyjä apuvälineitä . . . . .	11
4.3.1	3N-MCTS . . . . .	11
4.3.2	Expert knowledge aided neural networks . . . . .	16
<b>5</b>	<b>Lääkkeiden kehitys tulevaisuudessa</b>	<b>18</b>
	<b>Lähteet</b>	<b>19</b>



# 1 Johdanto

Uusien lääkkeiden tuottaminen on pitkä ja kallis prosessi. Tähän kuuluu useita vaiheita ja eri vaiheet vievät vaihtelevan määrän rahaa ja aikaa. Vaiheet ovat sairauden aiheuttajan tunnistaminen, tähän vaikuttavan lääkkeen tunnistaminen, lääkkeen optimointi, lääkkeen ominaisuuksien analysointi ja kliiniset testit. Vaiheiden jälkeen lääke joko hyväksytään myyntiin tai ei. Vaiheet ovat tavallisesti vieneet 10–12 vuotta ja kaikkien vaiheiden jälkeen työn hinnaksi on voinut muodostua 1–3 miljardia dollaria [5].

Vaiheiden pitkän keston ja suuren hinnan takia tutkijat ja lääkefirmat ovatkin alkaneet tutkia mahdollisia keinoja, jotka nopeuttaisivat tai halventaisivat tätä lääkkeen kehityksen prosessia. Koneoppimismallit ovat nousseet houkuttelevaksi vaihtoehdoksi, joka voisi nopeuttaa tätä prosessia. Lääkefirmat ovatkin alkaneet selvittää, kuinka eri koneoppimisen malleja voidaan hyödyntää lääketutkimuksessa nopeuttamaan suurimpia pullonkauloja [5]. Viimeisen kymmenen vuoden aikana saatavilla olevan laadukkaan datan määrä on kasvanut merkittävästi ja on kehitetty uusia tehokkaampia koneoppimismalleja, joita voidaan hyödyntää lääketutkimuksessa. Eri mallit voivat esimerkiksi karsia kaikista harkinnasta olevista lääkkeistä vain lupaavimmat kandidaatit, joilla on mahdollisuus päästä testeistä läpi tuotantoon. Koneoppimismalleja voidaan myös hyödyntää aivan uusien lääkeaineiden etsinnässä. Tähän tarkoitukseen kehitetyt koneoppimismallit kykenevät etsimään lääkkeitä, joilla on halutut sekä lääkkeelliset että fysikaaliset ominaisuudet [11]. Kehitetyt mallit ovatkin näyttäneet, että koneoppimismallit ovat tehokkaita työkaluja, joita voidaan hyödyntää kaikissa lääketutkimuksen prosessin vaiheissa.

Tässä tektissä paneudutaan syvemmin koneoppimismalleihin, joita käytetään uusien lääkkeiden tunnistamiseen ja näiden tunnistettujen lääkkeiden syntetisoinnin suunnitteluun.

Ensimmäisessä kappaleessa tutustutaan hieman eri datatyyppeihin, joita käytetään koneoppimismallien kouluttamisessa. Näiden ymmärtäminen auttaa hahmottamaan myöhemmin tekstissä selitettävien eri mallien toimintaa.

Toisessa kappaleessa käydään läpi, kuinka uusia lääkkeitä löydetään. Kappaleessa kerrotaan virtualisesta seulonnasta (Virtual screening), mitä sillä tarkoitetaan ja kuinka sitä käytetään hyväksi. Lisäksi toisessa kappaleessa käydään läpi kahden eri koneoppimismallin toiminta. Nämä koneoppimismallit keskittyvät uusien lääkkeiden löytämiseen, mutta niiden perustoimintaperiaatteet eroavat toisistaan.

Kolmannessa kappaleessa käsitellään yhdisteiden syntetisointia. Kappaleessa kerrotaan, mikä tekee yhdisteiden syntetisoinnista hankalaa ja mitä ratkaisuja tähän on kehitetty. Kappale myös kertoo retrosyntetisoinnista, CASP -menetelmistä ja kuinka nämä liittyvät toisiinsa. Lisäksi kappaleessa myös käyddän läpi kahden eri koneoppimismallin toiminta, jotka on kehitetty auttamaan eri yhdisteiden syntetisoinnissa.

Neljäs kappale käsittelee, kuinka koneoppimista tullaan käyttämään tulevaisuuden lääkekehityksessä ja tutkimisessa. Kappale kertoo, kuinka koneoppiminen tulee enenemissä määrin muuttamaan lääkekehityksen prosessia.



## 2 Kemiallinen data

Tässä kappaleessa käydään pintapuolisesti läpi, minkälaista dataa käytetään mallien kouluttamiseen. Kappaleessa keskitytään datatyyppeihin, joita käytetään malleissa, joiden toimintaa tullaan avaamaan tämän tekstin muissa osissa. Mallit, joiden toimintaa avataan, käyttävät datatyyppejä, joilla kuvataan pieniä yhdisteitä. Tämän takia kappaleessa keskitytään pieniäyhdisteitä kuvaaviin datatyyppeihin. Kappaleessa käyddän läpi SMILES ja SMARTS dattyyypit, mihin niitä käytetään ja kuinka ne eroavat toisistan. Lisäksi kappaleessa kerrotaan ominaisuusvektoreista ja kuinka niitä käytetään ja missä tilanteissa. oleelliset asiat datasta Mitä datasta halutaan tietää?

### 2.1 SMILES, SMARTS ja SMIRKS

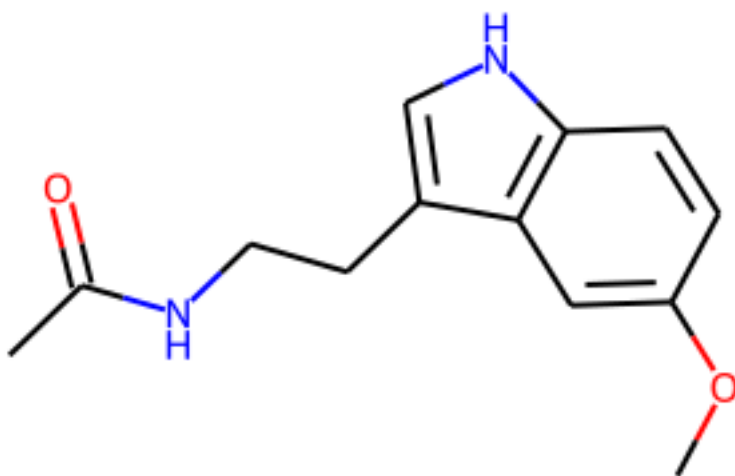
Mitä on? Mitä kertoo? Miten käytetään malleissa? (esikäsittely)

SMILES (Simplified Molecular Input Line Entry System) on David Weiningerin vuonna 1988 kehittämä tapa kuvata molekyylejä ja yksinkertaisia yhdisteitä [**WeiningerSMILES**]. Se kuvaa molekyylin yhdellä rivillä käyttäen ASCII -merkistöä. Tämän takia se on tietokoneen nopea lukea ja kemistin helppo ymmärtää.

SMARTS on puolestaan SMILES:ista jatkettu kieli, joka mahdollistaa yhdisteiden sisäisten rakenteiden määrittämisen. Tämän takia kaikki SMILES -merkkijonoiksi kelpaavat merkkijonot ovat myös kelpoja SMARTS -merkkijonoja.

SMIRKS on edelleen jatkokehitetty tapa kuvata yhdisteiden välisiä reaktioita ASCII -merkistöllä.

### 2.2 Ominaisuusvektorit



CC(=O)NCCC1=CNC2=C1C=C(C=C2)OC

**Kuva 2.1:** Kuvassa melatoniini molekyyli, kuinka se kuvataan graaffina ja SMILES -merkijonona. Kuva on tehty käyttämällä RDKit -ohjelmistoa.

[9]

## 3 Uusien lääkkeiden löytäminen

Tässä kappaleessa kerrotaan, mitä asioita liittyy lääkkeiden löytämiseen. Lisäksi kappaleessa kerrotaan, mitä on virtuaalinen seuronta ja kuinka sitä käytetään lääkkeiden löytämisessä. Kappaleessa myös käydään läpi kahden eri koneoppimismallin toimintamalli. Toinen malleista löytää uusia lääkkeitä muokkaamalla jo tunnettuja lääkkeitä. Toinen puolestaan luo uusia lääkkeitä perustuen ominaisuusvektoreihin.

### 3.1 Lääkkeiden löytäminen

Yksi ensimmäisistä lääketutkimuksen prosessin osa-alueista on uusien lääkeyhdisteiden löytäminen joko uusiin tai jo tunnettuihin tauteihin [5]. Lääkkeiden löytäminen on kuitenkin ollut tavallisesti hidas prosessi ja uuden toimivan yhdisteen löytäminen on kestänyt kahdesta kolmeen vuotta.

Viimeisimmän kymmenen vuoden aikana saatavilla olevan datan määrä on kasvannut merkittävästi kehitettyjen tietopankkien takia [5]. Näitä ovat esimerkiksi PubChem ja ChEMBL. Lisääntynyt datan määrä on täten mahdollistanut lääkkeiden löytämisen nopeuttamisen koneoppimismallien avulla.

Popovan et al. tutkimusryhmä on kehittänyt mallin, joka ehdottaa uutta yhdistettä perustuen mallin syötteenä saamaan ominaisuusvektoriin [7]. Harel et al. puolestaan ovat kehittäneet mallin, joka luo uusia yhdisteitä, jotka perustuvat syötteenä annettuun prototyyppi-yhdisteeseen [6]. Läkkeitä myös kehitetään muokkaamalla jo tunnettua lääkeyhdistettä jolla on jo osittain halutut ominaisuudet [6].

### 3.2 Virtuaalinen seulonta

Kemiallisten yhdisteiden avaruudella tarkoitetaan kaikkien uniikkien yhdisteiden lukumäärää. Erilaisten kemiallisten yhdisteiden avaruus on suuri. On arvioitu, että erilaisia kemiallisia yhdisteitä, jotka voivat esiintyä huoneenlämmössä ja nesteessä, voi olla välillä  $10^{18} - 10^{180}$  [10]. Yhdisteet, jotka täyttävät lääkkeitä vaaditut kriteerit, on puolestaan arvioitu olevan noin  $10^{60}$  [10]. Erillaisten yhdisteiden suuri lukumäärä itsessään esittää

tarpeen tehokkaille algoritmeille ja menetelmille, jotka auttavat karsimaan tästä suuresta määrästä kemiallisia yhdisteitä vain lupaavimmat.

Virtuaalinen seulonta (Virtual Screening, VS) on suosittu lähestymistapa uusien lääkkeiden löytämiseksi. VS käsittää joukon menetelmiä, joissa tietokoneita hyväksi käyttäen karsitaan yhdisteiden avaruudesta vain tietyt kriteerit täyttävät yhdisteet, joita voidaan jatkotutkia ja kehittää lääkkeiksi. VS menetelmillä tarkoitetaan yleisesti prosesseja, joissa käydään läpi suuria tietokantoja dataa, jotta löydetään haluttu yhdiste [10].

### 3.3 Kehitettyjä koneoppimismalleja

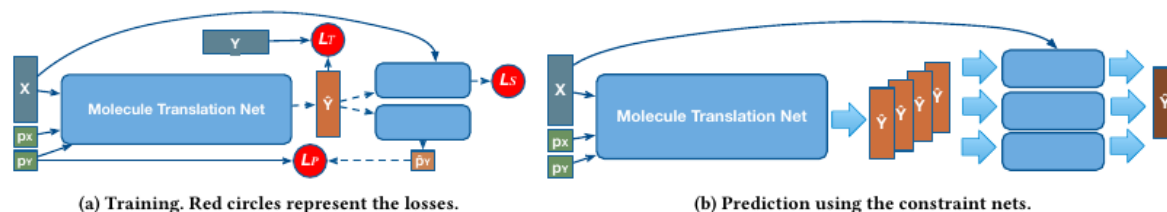
#### 3.3.1 Controlled molecule generator, CMG

CMG (Controlled molecule generator) on koneoppimismalli, joka etsii uusia molekyylejä, jotka perustuvat syötteenä annettuun molekyyliin ja joilla on ennalta määritellyt halutut ominaisuudet [9]. Se eroaa muista kehitetyistä malleista siten, että se pystyy optimoimaan annetun molekyylin useampaa ominaisuutta. Edelliset kehitetyt mallit ovat pystyneet optimoimaan vain yhtä molekyylin ominaisuutta.

CMG:n kehittäjät lähestyvät molekyylin ominaisuuksien optimointiongelmää merkkijonon käännös/luontiongelmana. CMG:lle opetetaan, kuinka syötteenä annettu molekyyli-merkkijono käännetään molekyyli-merkkijonoksi, jolla on lähimpänä haluttuja ominaisuuksia olevat ominaisuudet. CMG tulkitsee ensin annetut merkkijonot hyödyntäen DN:ää (deep network) jonka jälkeen se luo uusia molekyyliyhdisteitä hyödyntäen tätä tulkintaa ja haluttua ominaisuusvektoria. Koska ominaisuudet annetaan vektorina, niin CMG pystyy optimoimaan useampaa ominaisuutta. CMG lisäksi hyödyntää ennalta koulutettuja rajoitteverkkoja (constraint network, CN), jolloin vältetään luomasta mahdottomia yhdisteitä. CMG käyttää näitä verkkoja hyödyksi käyttämällä muokattua Beam Search -algoritmia. (Taustatietoa muista aiheista/malleista, joita on hyödynnetty CMG:ssä...)

CMG perusmalli on samanlainen verrattaen aikaisemmin kehitettyyn The Transformer -malliin (Molecule Translation Network, MTN) [12]. CMG:tä on laajennettu tästä lisäämällä siihen tietoa molekyylien ominaisuuksista ja kaksi CN:ää. Nämä CN:ät ovat ominaisuuksien ennustamiseen tarkoitettu verkko (property prediction network, PPN) ja samanlaisuuksien ennustamiseen tarkoitettu verkko (similarity prediction network, SPN).

MTN eroaa Transformer -mallista kahdella tavalla. Toisin kuin Transformer, joka käsitte-



Kuva 3.1: CMG:n toiminta periaate kuvattuna

[9]

lee sanoja ja niistä muodostettuja lauseita, MTN käsittelee yksittäisiä merkkejä ja niistä muodostettuja molekyylejä. Lisäksi MTN:än piilotettuun kerrokseen on lisätty tietoa kemiallisista ominaisuuksista. MTN:än kustannusfunktio on muotoa

$$\mathcal{L}(\theta_T; X, p_x, p_y) = -\frac{1}{N} \frac{1}{M} \sum_{n \in N} \sum_{i \in M} \sum_{v \in V} y_{v,j,n} \cdot \log(\hat{y}_{v,j,n}).$$

PPN ottaa syötteenä ennustetun molekyylin merkkijonon ( $y_i$ ). Tämä merkkijono muunnetaan piilotetuiksi vektoreiksi (Parempi termi?) hyödyntäen Long short-term memory (LSTM) -kerrosta. LSTM muodostaa vektoreita oikealta vasemmalle ja vasemmalta oikealle suunnassa, ja näistä vektoreista valitaan molempien suuntien viimeiset vektorit. Nämä vektorit yhdistetään ja yhdistevektori syötetään täysin yhdistetylle neuroverkolle. Tämä verkko sisältää kaksi piilotettua tasoa.

SMP ottaa syötteenä ennustetun molekyylin merkkijono ( $y_i$ ) ja alkuperäisen molekyylin merkkijonon ( $x_i$ ). Nämä merkkijonot syötetään kahdelle eri LSTM -tasolle, toinen käsittelee alkuperäisen molekyylin merkkijonot ja toinen ennustetun molekyylin merkkijonot. Nämä LSTM -kerrokset toimivat samalla periaatteella kuin PPN:ässä ja palauttavat näin ollen neljä eri vektoria. Nämä vektorit yhdistetään ja tämä yhdiste vektori annetaan täysin yhdistetylle verkolle, joka sisältää kaksi piilotettua tasoa.

Data, jolla, mallit koulutetaan, on peräisin ZINC ja DRD2 dataseiteistä. (Kuka hallinnoi?) Data sisältää kaiken kaikkiaan 257 565 molekyylä, joista luodaan pareja. Parit muodostetaan siten, että kahden molekyylin välinen samanlaisuus luku on yli 0.4. (Miten lasketaan?) Näitä pareja ( $X, Y$ ) muodostetaan kaiken kaikkiaan 10 827 615 kappaletta. Lisäksi kaikille molekyyleille lasketaan niiden eri kemiallisten ominaisuuksien arvot, jotka ovat PlogP, QED ja DRD2. Nämä kuvaavat eri haluttuja ominaisuuksia, joita lääkkeitä halutaan.

PPN koulutetaan käyttäen 257 565 eri molekyylä. nämä molekyylit jaetaan satunnaisaraisella valinnalla koulutus -ja testiryhmiin suhteella 8:2.

SPN kouluttamiseen käytetään osajoukkoa 10 827 615 eri parista. Tästä määrästä valitaan kymmenen prosentin osajoukko, joka on kaiken kaikkiaan 997 773 molekyylin suuruinen. Tämä joukko jaettiin samalla tavalla koulutus ja testi ryhmiin kuin PPN:ssä.

### 3.3.2 Prototyyppiin perustuva lääkekehitys

Yksi tapa luoda uusia lääkkeitä on muokata jo saatavilla olevia lääkkeitä tai lääkkeenkaltaisia yhdisteitä [6]. Yhdistettä, jota muokataan tällaisessa lääkehityksessä, kutsutaan prototyyppiksi. Tämä lähestymistapa tavallisesti parantaa prototyypin toimintaa. Prototyypin toimintaa voidaan tehostaa, sen sivuvaikutuksia voidaan lieventää ja se voidaan muokata toimimaan henkilöille, joille aikaisempi versio ei toiminut.

Tätä toteuttamaan on kehitetty CDN-VAE. Se on koneoppimismalli, jonka toimintamalli perustuu VAE (Variational Auto Encoder) arkkitehtuuriin. CDN-VAE ensin

### 3.3.3 CDN-VAE

lähde [6]

## 4 Uusien lääkkeiden syntetisointi

Tässä kappaleessa käydään läpi, miten lääkkeiden syntetisoinnin suunnittelua voidaan tehdä koneoppimismallien avulla. Kappale keskittyy syntetisointiin retrosyntetisoinnin avulla. Kappaleessa kerrotaan, mikä tekee yhdisteen retrosyntetisoinnista vaikeaa ja kuinka sitä voidaan tehostaa eri koneoppimismallien avulla. Kappaleessa käydään läpi kahden koneoppimismallin toimintaa, jotka suorittavat halutun yhdisteen retrosyntetisointia.

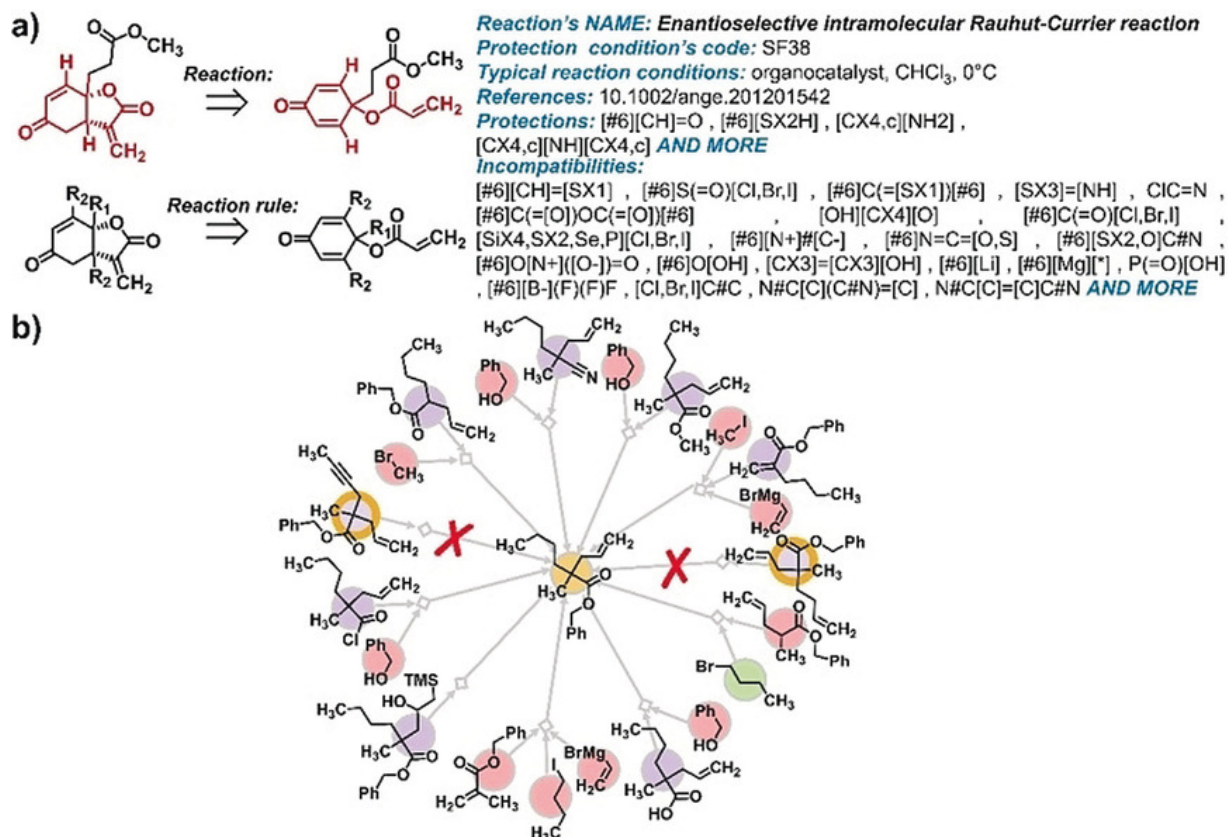
### 4.1 Yhdisteiden syntetisointi

Yhdisteen syntetisoinnin suunnittelulla tarkoitetaan prosessia, jossa määritellään, kuinka haluttu yhdiste voidaan tuottaa synteettisesti saatavilla olevista lähtöaineista [4]. Retrosynteesianalyysillä tarkoitetaan puolestaan menetelmää, jonka avulla löydetään halutun yhdisteen tuottamiseen tarvittavat lähtöaineet. Retrosynteesi toimii siis toiseen suuntaan kuin syntetisointi. Retrosynteesissä yhdiste pilkotaan rekursiivisesti pienempiin lähtöaineisiin kunnes jäljellä on vain saatavilla olevia lähtöaineita.

Tavallisesti yhdisteen retrosyntetisointi on vaatinut suorittavalta kemistiltä usean vuoden kokemusta ja tietoa saatavilla olevista lähtöaineista ja eri reaktioista. Tätä on pyritty automatisoimaan eri CASP -menetelmien avulla (Computer-Aided Synthesis Planning). Ensimmäiset CASP -menetelmät perustuivat heuristisiin algoritmeihin, joissa kemistit käsin koodasivat, miten eri lähtöaineet reagoivat keskenään ja mikä on reaktion lopputuote. Tämä on kuitenkin osoittautunut toivottomaksi yritykseksi massiivisen datan määrän takia.

Kehitys koneoppimismenetelmissä on kuitenkin tarjonnut uuden lähestymistavan CASP -menetelmien kehitykseen. Sen sijaan, että kemistit loisivat heuristisia malleja, niin uudet koneoppimismallit koulutetaan saatavilla olevan datan avulla. Tämä on todettu merkittävästi enemmän toteutettavaksi lähestymistavaksi.

Koneoppimismallien käyttö ja koulutus ei ole kuitenkaan täysin ongelmaton lähestymistapa myöskään. Ongelmaan liittyen dataa ei välttämättä ole saatavilla ja datan hankkiminen voi olla kallis operaatio. Tätä varten on kehitetty tietopankkeja, jotka sisältävät massiivisia määriä dataa tietystä aiheesta, esim. Reaxys kemiallisista reaktioista.



**Kuva 4.1:** (a) esimerkki reaktiosäännöstä, (b) esimerkki mahdollisista reaktioista, kuinka voidaan luoda haluttu yhdiste (keskellä). Kuvaa on yksinkertaistettu ja se sisältää vain yksitoista mahdollista reaktiota, jotka tuottavat halutun yhdisteen. Värikoodaukset tarkoittavat: Keltainen — kohde yhdiste, Punainen — saatavilla oleva yhdiste, Vihreä — kirjallisuudesta tunnettu yhdiste, liila — tuntematon yhdiste

[2]

## 4.2 Lääkkeen retrosyntetisoinnin haastavuus

Retrosyntetisoinnin tekee hankalaksi fakta, että yhdiste voidaan muodostaa sadoilla tai tuhansilla eri tavoilla. Tämä ongelma toistuu rekursiivisesti, kun yhdiste pilkotaan yhdisteisiin, jotka keskenään reagoiessa muodostavat alkuperäisen yhdisteen. Pienille ja yksinkertaisille yhdisteille tämä vaihtoehtoavaruus on pienempi, mutta yhdisteen koon kasvaessa eri tapojen määrä muodostaa haluttu yhdiste kasvaa eksponentiaalisesti.

Tämän takia tarve tätä prosessia yleistäville koneoppimismalleille on suuri. Miksi lääkkeiden retrosyntetisointi on hankalaa [3, 1]?



## 4.3 Kehitettyjä apuvälineitä

### 4.3.1 3N-MCTS

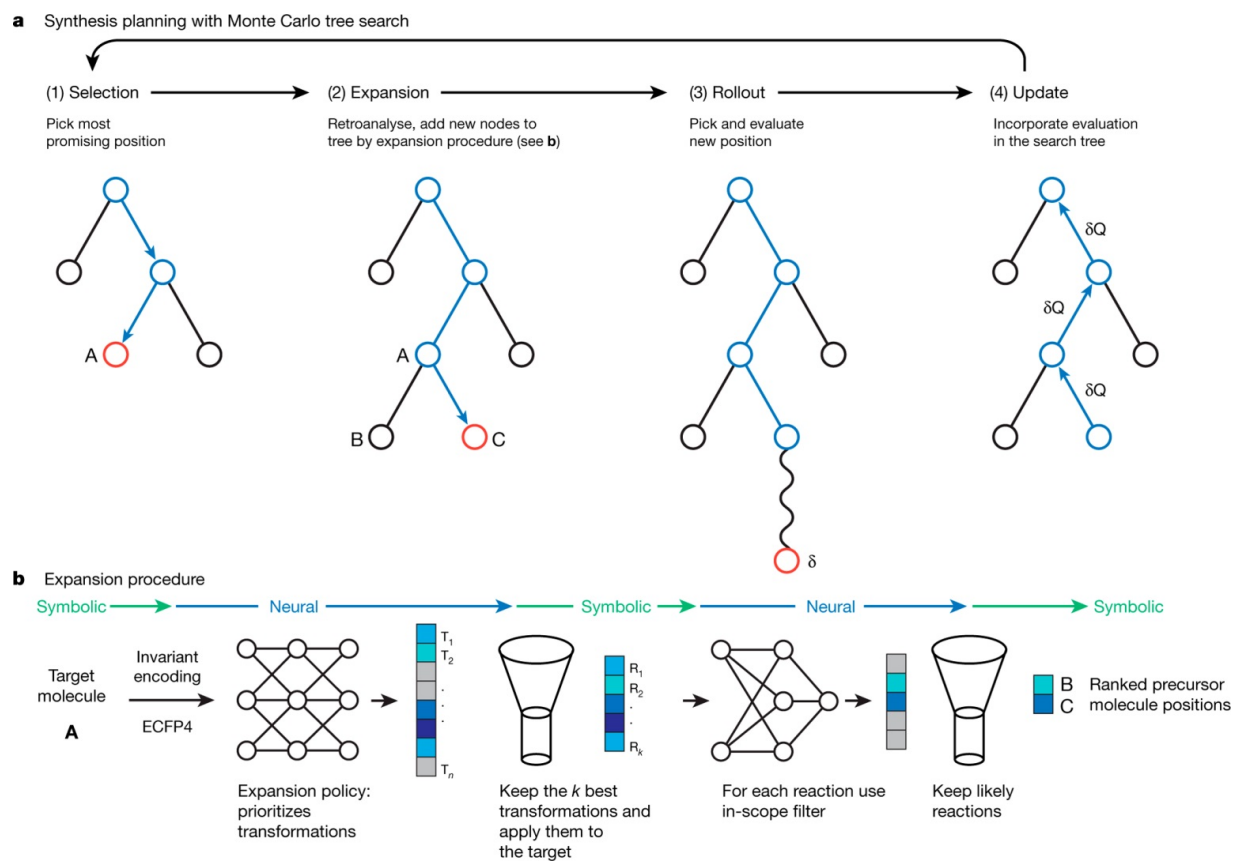
3N-MCTS on koneoppimismalli, jonka on kehittänyt Marwin et al; joka etsii retrosynteesipolkuja yksinkertaisempiin ja saatavilla oleviin lähtöaineisiin [8]. Kun retrosynteesipolku on varmennettu ja todettu toimivaksi, niin syötteenä annettu yhdiste on mahdollista syntetisoida laboratoriossa. 3N-MCTS koostuu kolmesta eri koneoppimismallista ja Monte Carlo -puuhakualgoritmista (**Monte Carlo Tree Search, MCTS**). Neuroverkot on koulutettu avustamaan puuhakualgoritmia etenemään fiksuimpaan suuntaan, kun haku algoritmi etsii syntetisointipolkuja ja tarkistamaan, onko ehdotettu reaktio mahdollinen kyseisellä molekyylillä.

Neuroverkot ovat hakupuun laajentumisen suuntaa ohjaava verkko (**Expansion policy network, EPN**), MCTS:än rollout toimintoa tukeva Rollout -verkko (**Rollout policy network, RPN**) ja verkko, joka tarkistaa, onko syntetisointi polku toteutettavissa (**In-scope filter network, IFN**).

Data, jolla neuroverkot koulutetaan, on peräisin Reaxys -tietokannasta. Reaxysin omistaa Elsevier kustantamo. Reaxys -tietokannan sisältämä data koostuu muunnossäännöistä (**transformation rule**), jotka kertovat, mitkä lähtöaineet reagoivat keskenään, mikä reaktio on kysessä ja mikä on reaktion tuote. Näitä sääntöjä käytetään mallien kouluttamiseen. Reaxys sisältää yli 12.4 miljoonaa sääntöä. Mallien kouluttamiseen käytetyt säännöt sisältävät vain yksivaiheisia kemiallisia reaktioita ja reaktiossa on mukana vain yhdestä kolmeen lähtötuotetta. Eri mallien kouluttamiseen käytettiin eri kriteerein suodatettua dataa tietokannasta. Kaikki koneoppimismallit koulutettiin käyttämällä ADAM -optimoijaa. ADAM -optimoija hyödyntää AdaGrad:in ja RMSProp:in toiminnallisuutta [kingma2017adam].

### 3N-MCTS toimintamalli

3N-MCTS:ssä IFN ja EPN on yhdistetty toimimaan yhdessä. Tutkittaessa puun tilaa  $S_i$  (selitä Si vaihe) jokainen molekyyli syötetään EPN:älle ja se tulostaa, mitkä säännöt voivat muodostaa annetun yhdisteen ja näin ollen myös mitkä lähtöaineet voivat muodostaa annetun yhdisteen. Nämä säännöt syötetään IFN:älle, joka suodattaa valituista reaktioista toteutettavissa olevat. Tämän jälkeen algoritmista iteroidaan neljää vaihetta,



Kuva 4.2: 3N-MCTS:än toiminta periaate kuvattuna

[8]

jotka muodostavat lopullisen puun.

(1) Ensimmäisessä vaiheessa algoritmi valitsee seuraavan lupaavimman tilan puusta kunnes puun lehti on saavutettu. Jos lehdessä käydään ensimmäisen kerran valintavaiheen aikana, niin lehti arvostellaan simuloimalla hakualgoritmia  $d$  askelta eteenpäin samalla muodostaen synteesipolkua (rollout). Jos lehdessä käydään useamman kuin yhden kerran valintavaiheen aikana, niin mahdolliset säännöt, jotka muodostavat lehden, tutkitaan ja lisätään lehden lapsiksi (expansion).

(2) Toisessa vaiheessa lupaavien tilojen lapset tutkitaan. Tällöin etsitään lupaavimmat säännöt, jotka muodostavat kyseisessä tilassa olevan yhdisteen.

(3) Kolmannessa vaiheessa tarkistetaan lehden tila. Jos lehti on ‘todistetusti toimiva’, niin algoritmi palauttaa luvun suuremman kuin yksi, jolloin lehteä suositellaan käytettävän synteesipolussa. Muussa tapauksessa lehdelle suoritetaan rollout, jolloin RPN antaa rekursiivisesti uusia sääntöjä niin kauan, kunnes lehti on pilkottu lähtöaineisiin tai kunnes suurin sallittu syvyys  $d$  on saavutettu.

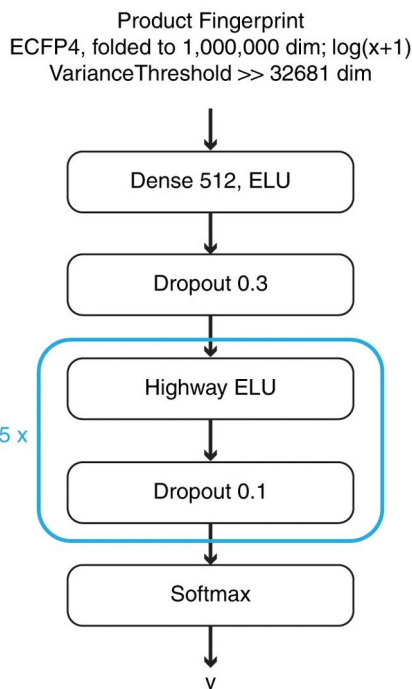
(4) Viimeisessä vaiheessa lehtien arvot päivitetään. Jos lähtöaineet löydetään rolloutin aikana, niin lehti saa palkonnoksi arvon 1. Jos Kaikkia lähtöaineita ei löydetty, niin lehdelle annetaan osittainen palkinto. Jos yhtään lähtöainetta ei löytynyt, niin lehti saa arvon -1.

Saattaa kuitenkin olla, että synteesi polkua ei voida luoda. Joko synteesinpolun tutkimiseen menee liian kauan aikaa tai synteesi polku sisältää liian monta vaihetta yhdisteen syntetisoimiseen.

### Expansion policy network

EPN muodostuu täysin kytketystä tasosta ja viidestä highway -tasosta. Molemmissa tasoissa on käytetty aktivointifunktiona ELU -funktioita (**exponential linear unit**). Täysin kytketyssä tasossa suoritetaan dropout, jonka jälkeen vain 0.3 ensimmäisen tason solmujen tuloksista otetaan huomioon seuraavalla tasolla. Tämän jälkeen jokaisen highway -tason jälkeen suoritetaan 0.1 dropout. EPN:än viimeinen taso on softmax, joka palauttaa todennäköisyysjakauman. Tämä jakauma kertoo kuinka todennäköisesti muunnossäntö  $a$  muodostaa tuotteen  $m$ .

EPN:än kouluttamiseen valittiin datasta vain reaktiokeskus. EPN:än datasta suodatettiin pois sellaiset säännöt, jotka ilmenivät datassa alle kolme kertaa ennen vuotta 2015. Lopullinen reaktiomäärä, jolla EPN koulutettiin, oli 301 671 reaktiota.

**Expansion Policy****Kuva 4.3:** Kuvassa expansion policy -neuroverkon rakenne.

[8]

**Rollout policy network**

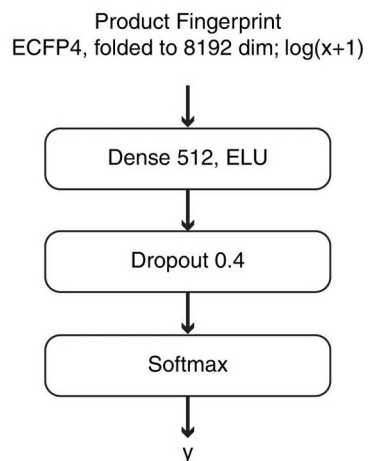
RPN muodostuu yhdestä täysin kytketystä tasosta. Samalla tavalla kuin EPN, RPN käyttää myös ELU -funktiota aktivointifunktiona. RPN:än viimeinen taso on myös softmax, joka palauttaa todennäköisyysjakauman.

RPN:än kouluttamiseen valittiin datasta vain reaktiossa muuttuneet atomit ja liitokset (reaktiokeskus) ja lähimmät vierekkäiset atomit. Datasta suodatettiin pois sellaiset säännöt, jotka ilmaantuivat alle 50 kertaa ennen vuotta 2015. Lopullinen reaktiomäärä, jolla RPN koulutettiin, oli 17 134 reaktiota.

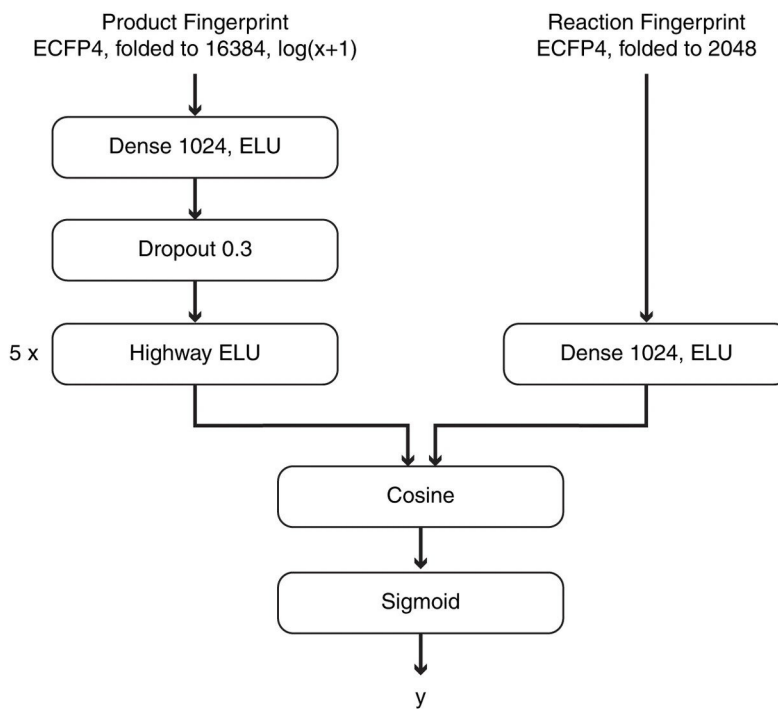
**In-scope filter network**

3N-MCTS:ässä IFN toimii binääriluokittelijana. Se kertoo, onko sille annettu yhdiste mahdollista muodostaa sille annetulla muunnossäännöllä.

IFN neuroverkko muodostuu kahdesta haarasta. Toinen haara saa syötteenään muodostettavan yhdisteen tunnisteen ja toinen haara saa muunnossäännön tunnisteen.

**Rollout Policy****Kuva 4.4:** Kuvassa Rollout policy -neuroverkon rakenne.

[8]

**In Scope Filter****Kuva 4.5:** Kuvassa In scope filter -neuroverkon rakenne.

[8]

Haara, joka saa syöteenä yhdisteen tunnistein, muodostuu yhdestä täysin kytketystä tasosta jonka jälkeen suoritetaan 0.3 dropout. Tämän jälkeen data kulkee viiden highway-tason läpi. Sekä täysin kytketty että highway -tasot käyttävät ELU -funktiota aktivointifunktiona.

Haara, joka saa syöteenään muunnossäännön tunnistein, muodostuu vain yhdestä täysin kytketystä tasosta. Tämä taso käyttää myös ELU -aktivointifunktiota.

Näiden haarojen jälkeen saaduista tuloksista lasketaan niiden kosinismankaltaisuus. Tämä syötetään sitten sigmoid -funktiolle, jonka tulos kertoo, pystyykö kyseinen muunnossääntö muodostamaan halutun yhdisteen.

IFN koulutetaan sekä onnistuneiden että epäonnistuiden sääntöjen avulla. Koska epäonnistuneita sääntöjä ei tallenneta tietokantaan, niin kyseinen data generoidaan. Data generoidaan siten, että jos reaktiossa



lähtöaineet A ja B muodostavat reaktiossa lopputuotteen C, niin lopputuotteita D, E, F, jne. ei muodostu (voisi selittää syvemmin). IFN kouluttamista varten luotiin 100 miljoonaa epäonnistunutta reaktiota ja 10 miljoonaa testaamista varten.

### 3N-MCTS suorituskyky

#### 4.3.2 Expert knowledge aided neural networks

On myös yleistä, että retrosynteesi polkuja suunnitelevat mallit hyödyntävät myös heuristiikka [2]. Badowski et al ryhmä on kehittänyt mallin, joka suunnittelee retrosynteesi polkuja hyödyntämällä neuroverkkoja ja ammattikemistien muodostamia heuristisia malleja suunnitellessaan retrosynteesi polkua.

ICHO:n (Instytut Chemii Organicznej) kouluttamiseen käyettiin dataa 1.4 miljoonasta reaktiota ja niiden lopputuotteista. Data saadaan julkaisuista artikkeleista ja patenteista. Reaktiolta kuitenkin vaaditaan, että ainakin yksi reaktio per lopputuote löytyy myös Chematican (taustatietoa) expert-coded säännöstöstä. Lisäksi datasta suodatettiin pois suojaryhmien säännöt.

ICHO:sta kehitettiin myös malli, jonka kouluttamisessa käytettiin lisäksi vektoreita, jotka sisälsivät dataa kemiallisesti intuitiivisista reaktio piirteistä, joita on käytetty aikaisemmissa tutkimuksissa. Tätä mallia kutsutaan ICHO+ malliksi, ja sitä käytetään vertailemaan ICHO mallin suoriutumista.

Kehitetyssä ICHO mallissa on kaksi tärkeää seikkaa.

(1) Se laskee reaktioiden ilmaantumistodennäköisyyttä uudelleen perustaen kuinka usein ne ilmaantuvat ammattilaisten käyttämissä reaktiossa. Tämä laskenta tapahtuu mallin koulutusvaiheessa. Käytännössä tämä tarkoittaa, että koulutusvaiheessa mallilla on tieto, kuinka monta kertaa tietty reaktiosääntö ilmeni datassa ja kuinka monta kertaa kyseistä reaktiota käytettiin luomaan jokin lopputuote. Malli laskee näiden välisen suhteen, jolloin malli pystyy määrittämään, kuinka usein kyseistä reaktioita kannattaa käyttää. Jos esimerkiksi jokin reaktio on mainittu datassa kymmenen kertaa ja samassa datassa kyseistä reaktiota käytetään kymmenen kertaa luomaan jokin lopputuote, niin malli käyttää tätä reaktiota, koska se luokitellaan ‘Helpoksi ja varmaksi toteuttaa’. Jos suhde taas on pieni, niin reaktio luokitellaan ‘vaikeaksi toteuttaa’.

(2) Se kykenee antamaan suuremman kuin nollatodennäköisyyden reaktiolle, joka ei ilmaantunut koulutusvaiheessa. Tämä sen takia, että mallille koulutetaan jatkuva funktio, joka antaa todennäköisyyden jokaiselle reaktiotyypille perustuen, kuinka reaktio muuttaa lopputuotteen lähtötuotteiksi. Eli jos jokin reaktion sormenjälki (Lisää tietoa, koska useassa mallissa käytetään) tai heuristinen kuvaus on samankaltainen toisen reaktion kanssa, joka ilmenee koulutusdatassa, niin reaktiolle voidaan antaa todennäköisyys sen käytölle.

ICHO mallia vertailtiin aikasiemmin mainitun 3N-MCTS mallin kanssa, jota kutsutaan artikkelissa SW mallina. Kyseisestä SW mallista kehitettiin myös heuristista dataa hyödyntävä malli SW+. Lisäksi luotiin SW malli, joka ei käy läpi kaikkia mahdollisia reaktioita, vaan valitsee säännöt vain niistä reaktioista, jotka eivät aiheuta konfliktia (selitys) ja jotka johtavat vain tiettyyn lopputuotteeseen. Tätä mallia kutsutaan SW2 malliksi ja täsät luodusta heuristisesta mallista käytetään nimeä SW2+ Lisäksi vertailumalliksi luodaan myös täysin heuristinen malli, joka arvioi reaktioita sen mukaan, kuinka paljon ne yksinkertaistavat lopputuotetta rakenteellisesti. Tämä malli suosii reaktioita, jotka pilkkovat molekyylin keskeltä, eli puolittavat sen. Tämä on haluttu lähestymistapa retrosynteisiin, koska se minimoi syntetisointi vaiheiden määrän.

## ICHO suorituskyky

Miten koneoppimista hyödynnetään tällä hetkellä lääkkeiden syntetisoinnissa [8, 6, 9]?

## 5 Lääkkeiden kehitys tulevaisuudessa

Vaikka koneoppimisen käyttökohteet ovat yleistyneet lääkekehityksessä, niin koneoppimisen lähestymistavoilla on edelleen potentiaalia kehittyä [3].

Yksi kehityksen kohteista on tehokkaampi datan tulkinta pienestä määrästä dataa. Paikoin kemian ja lääkekehityksen alalla on muihin koneoppimismallien käyttökohteisiin verrattaen vähän dataa saatavilla ja sitä on kallista tuottaa. Tämä vaatii tutkijoilta enemmän työtä, jotta tieteellisistä julkaisuista saatava data saadaan koneiden hyödyntämään muotoon. Toinen ratkaisu tähän on meta-oppimis (selitys) lähestymistapa. Uudet lähestymistavat kuten Neuro Turing kone ja matkija oppiminen mahdollistavatkin oppimisen vähästä määrästä dataa ja Bayes luokittelija on suoriutunut lähes ihmistasoisesti käyttämällä One-shot oppismallia.

Toinen merkittävä edistysaskel on kemiallisten yhdisteiden ja reaktioiden tehokkaampi esitysmalli. Tähän asti kemiaa on esitetty ihmisten ymmärtämässä muodossa, mutta tämä ei aina ole koneelle paras esitysmalli. Koneoppimismallit käyttävät tietoa molekyyleistä ja atomeista ja koneoppimismallit ovat niin hyviä kuin nämä kuvaukset. Hyvältä kuvaukselta vaaditaan, että sen avulla on yksinkertaista hankkia kohteen ominaisuudet ja sen tulee olla mahdollisimman pieni ulottuvuinen. Uusia kuvausmalleja on kehitetty ja uudet mallit ovat näyttäneet, että ne ovat tehokkaita, niin uusien kuvauksien kehittäminen jatkuu edelleen.

Kemiallista koneoppimista edistävä asia on myös kvanttilaskennan hyväksi käyttö. Kvanttikoneiden suuri laskuteho pystyy laskemaan uusien mallien koulutusaikaa merkittävästi.

Koneoppimista voidaan myös käyttää tulevaisuudessa mahdollisesti uusien tieteellisten lakien löytämisessä. Mutta vaikka eri koneoppimisesta kehitetyt mallit ovat ennalta arvattavia, niin ne ei kuitenkaan ole aina tulkittavia. Neuroverkko voi esimerkiksi oppia ideaalin kaaulain ( $pV = nRT$ ), mutta neuroverkon kaarien painojen muuttamien ymmärrettäväksi säännöksi ei ole yksinkertainen tehtävä. Voi olla, että koneoppimismalli pystyy huomaamaan datasta säännön, joka toistuu, mutta jos tutkijat eivät tiedä tai tunne kyseistä sääntöä, niin mallin tulkitseminen on lähes mahdotonta.



# Lähteet

- [1] A. F. de Almeida, R. Moreira ja T. Rodrigues. "Synthetic organic chemistry driven by artificial intelligence". eng. *Nature reviews. Chemistry* 3.10 (2019), s. 589–604. ISSN: 2397-3358.
- [2] T. Badowski, E. P. Gajewska, K. Molga ja B. A. Grzybowski. "Synergy Between Expert and Machine-Learning Approaches Allows for Improved Retrosynthetic Planning". *Angewandte Chemie International Edition* 59.2 (2020), s. 725–730. DOI: <https://doi.org/10.1002/anie.201912083>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.201912083>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201912083>.
- [3] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev ja A. Walsh. "Machine learning for molecular and materials science". eng. *Nature (London)* 559.7715 (2018), s. 547–555. ISSN: 0028-0836.
- [4] C. W. Coley, W. H. Green ja K. F. Jensen. "Machine Learning in Computer-Aided Synthesis Planning". eng. *Accounts of chemical research* 51.5 (2018), s. 1281–1289. ISSN: 0001-4842.
- [5] S. Ekins, A. C. Puhl, K. M. Zorn, T. R. Lane, D. P. Russo, J. J. Klein, A. J. Hickey ja A. M. Clark. "Exploiting machine learning for end-to-end drug discovery and development". eng. *Nature materials* 18.5 (2019), s. 435–441. ISSN: 1476-1122.
- [6] S. Harel ja K. Radinsky. "Accelerating Prototype-Based Drug Discovery Using Conditional Diversity Networks". Teoksessa: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, s. 331–339. ISBN: 9781450355520. DOI: 10.1145/3219819.3219882. URL: <https://doi-org.libproxy.helsinki.fi/10.1145/3219819.3219882>.
- [7] M. Popova, O. Isayev ja A. Tropsha. "Deep reinforcement learning for de novo drug design". eng. *Science advances* 4.7 (2018), eaap7885–eaap7885. ISSN: 2375-2548.
- [8] M. H. S. Segler, M. Preuss ja M. P. Waller. "Planning chemical syntheses with deep neural networks and symbolic AI". eng. *Nature (London)* 555.7698 (2018), s. 604–610. ISSN: 0028-0836.

- [9] B. Shin, S. Park, J. Bak ja J. C. Ho. "Controlled Molecule Generator for Optimizing Multiple Chemical Properties". Teoksessa: *Proceedings of the Conference on Health, Inference, and Learning*. CHIL '21. Virtual Event, USA: Association for Computing Machinery, 2021, s. 146–153. ISBN: 9781450383592. DOI: 10.1145/3450439.3451879. URL: <https://doi.org/10.1145/3450439.3451879>.
- [10] C. Sottriffer, R. Mannhold, H. Kubinyi ja G. Folkers. *Virtual Screening: Principles, Challenges, and Practical Guidelines*. eng. Vol. 48. Methods and principles in medicinal chemistry. Weinheim: John Wiley ja Sons, Incorporated, 2011. ISBN: 9783527326365.
- [11] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer ja S. Zhao. "Applications of machine learning in drug discovery and development". eng. *Nature reviews. Drug discovery* 18.6 (2019), s. 463–477. ISSN: 1474-1776.
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser ja I. Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].