# Database toolkit voor kanker varianten

## Situering

Binnen het Vlaamse ziekenhuisnetwerk geldt AZ Delta als een van de referentiecentra voor genetisch onderzoek, vooral binnen de applicaties van NIPT, oncologie en microbiologie (o.a. covid, griep, soa’s, e.a.). Het Bio-IT team dat verantwoordelijk is voor de dataverwerking van deze analyses voerde vorig jaar in totaal meer dan 25000 analyses uit.

Binnen de tak oncologie bestaat de dataverwerking uit het opzoeken waarom de tumor de eigenschappen heeft die het vertoont (diagnose), en het linken aan mogelijke behandelingen (type, dosering en duur van chemo-/radio-/immunotherapie). Het aantal te verwerken stalen is eerder beperkt (40a50/week), maar de interpretatie van deze resultaten is arbeidsintensief (3 man fulltime). Dit komt omdat verschillende databases geraadpleegd moeten worden online, of informatie inefficiënt bijgehouden wordt in Word of Excel.

De eerste stap van de dataverwerking is volledig geautomatiseerd. Hierbij wordt het genetisch materiaal van de tumor in kaart gebracht, en wordt deze vergeleken met wat we verwachten (een referentie sequentie). Bij de vergelijking worden alle afwijkende posities uit alle data gehaald. Deze afwijkende posities variëren sterk van patiënt tot patiënt. Bij sommige kanker types, zoals bij borstkanker, zijn er specifieke locaties die bij veel patiënten voorkomen (zogenaamde hotspots). Voor vele van deze posities is er informatie over de geadviseerde behandeling beschikbaar in het labo (in Word of Excel). Echter zijn er ook vaak posities die zelden voorkomen. Deze posities zijn vaak minder goed gedocumenteerd, en hebben nog extra opzoek werk nodig.

Er is duidelijk een nood aan een gestroomlijnd platform waar de informatie van gekende afwijkingen wordt bijgehouden, waar ook nieuwe informatie op een gemakkelijke manier toegevoegd kan worden.

## Opdracht

In dit project moet een database ontwikkeld worden waarbij een positie bijgehouden en geraadpleegd kan worden, samen met alle mogelijke informatie. Echter is het onmogelijk om op voorhand te zeggen welk soort van informatie en hoeveel dit zal zijn. Anderzijds is het ook nodig om informatie over de genen zelf bij te houden. Een positie kan in een of meerdere genen voorkomen.

Op deze database wordt een API ontwikkeld waardoor de database bevraagd kan worden naar de informatie van een bepaalde positie, of waarbij nieuwe informatie kan worden toegevoegd of geüpdatet. De API zal enkel naar data vragen die op een gestructureerde manier bijgehouden wordt.

Er is ook een web-gebaseerde interface nodig die gebruikers toelaat om posities op te vragen, nieuwe toe te voegen of aanpassingen te maken. Hierbij moet de data op een duidelijke manier getoond kunnen worden. Hierbij is belangrijk dat er ook kan gekeken worden wanneer de laatste update was. Indien dit langer dan x tijd geleden was, moet dit ook automatisch aangeduid kunnen worden, zodat het duidelijk is voor de gebruiker dat een review van deze positie nodig is.  
Tevens dient de gebruiker ook te kunnen doorklikken naar een pagina die gelinkt is aan het gen van deze positie. Omgekeerd moet van de gen pagina kunnen gegaan worden naar alle gekende posities. Voor de gen pagina is naast een overzicht van alle gekende posities ook nodig dat er informatie getoond, toegevoegd of gewijzigd kan worden. Deze informatie zal vooral vrije tekst zijn. Hierbij is het ook interessant om afbeeldingen te kunnen toevoegen.

## Vereisten

* Het Bio-IT team werkt op Linux servers en in een container systeem. Het ontwikkelen binnen containers valt buiten de scope van de opdracht. Echter zal de code uitgetest worden binnen een Fedora container in podman. Om zeker te zijn dat de code zal werken wordt aangeraden om binnen een Virtuele Machine te werken zoals bv VirtualBox ([www.virtualbox.org](http://www.virtualbox.org)), en hierbinnen Fedora te installeren (<https://getfedora.org/nl/>). Indien je extra tools installeert moet dit ook gedocumenteerd worden!
* De code mag geïmplementeerd worden in de programmeertaal naar keuze, maar de API moet raadpleegbaar zijn in Python, Groovy, Java en C++. Indien de voorkeur gaat naar ontwikkeling in .NET, moet rekening gehouden worden dat dit niet rechtstreeks op Linux kan lopen. Oplossing hiervoor zijn te vinden binnen mono (<https://www.mono-project.com/>) en GCC (<https://gcc.gnu.org/>).
* De data voor de positie waarop geselecteerd moet kunnen worden is: chromosoom, positie binnen dit chromosoom, de referentie base en de nieuwe geobserveerde base. Verdere data die hierbij samenhangt kan in vrije tekst bijgehouden worden (zoals referenties (URLs) naar artikels, tekst met beschrijving van effect, …). Tot slot zijn er nog gestructureerde waarden nodig. Deze zijn niet op voorhand allemaal definieerbaar, dus is het wenselijk dat uitbreiding mogelijk is zonder aanpassing van de code. Gestructureerde data kan zijn: effect, geneesmiddel, positie in het eiwit, ids van externe databases (die niet doorgelinkt moeten worden).
* Het opzoeken van een gen moet simpel kunnen door de volledige gen code in een zoekbalk te typen (een gencode is maximaal 7 karakters lang, dus dit vormt geen probleem voor de gebruiker). Het gen zelf zou een vrije tekst moeten kunnen bevatten, en afbeeldingen. Alle afbeeldingen mogen onderaan de pagina elkaar opvolgen. In dit geval is het aan de gebruiker om gemakkelijk te refereren. Een meerwaarde zou zijn moest er een structuur in de vrije tekst toegevoegd kunnen worden (denk aan simpele markdown die naar html vertaald wordt).
* Hint: mogelijk is het makkelijker om voor deze opdracht een combinatie uit te werken van een traditionele SQL database, met een noSQL variant, of met eenvoudige JSON files. Let wel op dat backups van de ontwikkelde opslagmethode eenvoudig en eenduidig moeten zijn.
* Meer duidelijke voorbeelden over hoe de data op zich eruit ziet zullen geleverd worden bij de start van het project.