Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Marlena Osipowicz
Nr albumu: 372240

Klasyfikacja pacjentów z chorobą Alzheimera na podstawie polimorfizmów DNA

Praca licencjacka
na kierunku BIOINFORMATYKA I BIOLOGIA SYSTEMÓW

Praca wykonana pod kierunkiem
dr Magdaleny Machnickiej
Instytut Informatyki

Wrzesień 2018

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora pracy

Streszczenie

W pracy przedstawiono proces budowy klasyfikatora pacjentów z chorobą Alzheimera, opierającego się na informacji o jednonukleotydowych polimorfizmach DNA (tak zwanych SNP-ach). Przeprowadzono również analizę uzyskanych wyników w celu ustalenia najistotniejszych miejsc w genomie, mogących mieć wpływ na rozwój choroby. Lepsze poznanie podstaw genetycznych choroby Alzheimera daje szanse na ustalenie dokładniejszych przyczyn choroby, co z kolei jest niezbędne do opracowania skutecznych metod jej leczenia.

Słowa kluczowe

choroba Alzheimera, klasyfikacja, algorytm Boruta, polimorfizmy DNA, SNP, drzewa decyzyjne, lasy losowe

Dziedzina pracy (kody wg programu Socrates-Erasmus)

11.3 Informatyka, nauki komputerowe

Tytuł pracy w języku angielskim

Classification of patients with Alzheimer's disease based on DNA polymorphisms

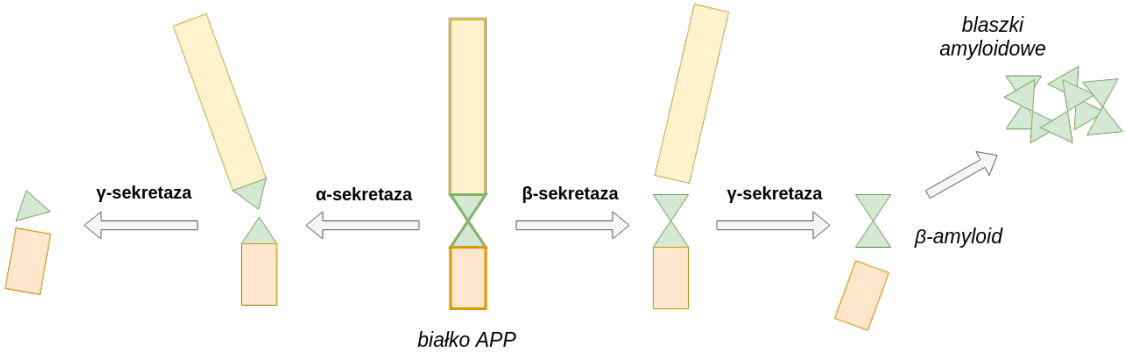
Spis treści

Rozdział 1

Wstęp

1.1. Choroba Alzheimera

Choroba Alzheimera (w skrócie AD - *Alzheimer's Disease*) jest chorobą neurodegeneracyjną, będącą najczęstszą przyczyną demencji [1]. Jest ona przykładem dolegliwości o złożonych podstawach genetycznych. Zmiany genetyczne powiązane z rozwojem AD mogą być zarówno dominujące i charakteryzujące się wysokim poziomem penetracji, rozumianej jako częstość ujawniania się w fenotypie cechy kodowanej przez dane loci, jak i recesywne oraz charakteryzujące się niską penetracją, ale względnie często występujące w populacji. Przykładem zmian pierwszego rodzaju mogą być mutacje w obrębie genów APP, PSEN1 i PSEN2, a drugiego rodzaju warianty genu SORL1 [2]. Mechanizmy stojące za procesem rozwoju choroby Alzheimera nie zostały w pełni poznane, jednak za kluczowe dla rozwoju choroby uznaje się procesy powstawania tak zwanych *blaszek amyloidowych* oraz *splątków neurofibrylarnych* [3].

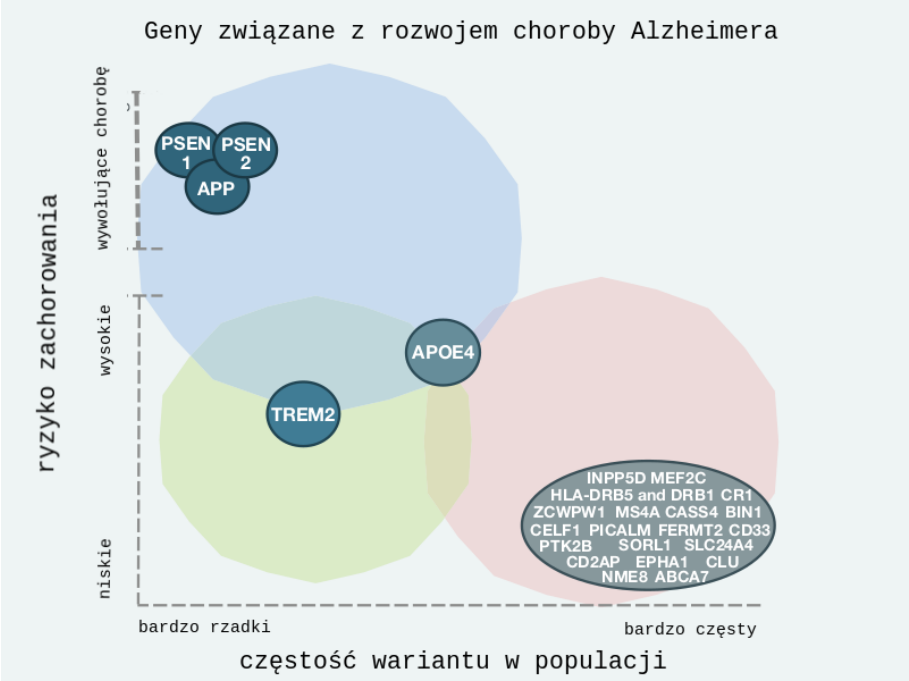


Rysunek 1.1: Schemat rozkładu białka APP zależnie od enzymów biorących w nim udział.

Uważa się, że jednym z najważniejszych czynników powodujących rozwój AD jest akumulacja peptydów β-amyloidowych (w skrócie Aβ) w mózgu. Powstają one podczas rozkładu białka APP (*Amyloid Precursor Protein*) przez enzymy z grupy sekretaz (α-, β- i γ- sekretazy). Białko APP jest białkiem błonowym, biorącym udział w procesie wzrostu i naprawy komórki nerwowej. Jego rozkład może przebiegać na wiele sposobów, zależnie od przeprowadzających go enzymów (Rys. 1.1). Od nich zależy, na jakie peptydy białko APP zostanie rozcięte. W organizmie osoby chorej na chorobę Alzheimera białko APP jest częściej niż normalnie

rozkładane do tak zwanego β-amyloidu 42 [4]. Peptyd ten ma tendencje do tworzenia złogów, co prowadzi do powstawania *blaszek amyloidowych*. Blaszki zalegają w przestrzeni międzykomórkowej, zaburzając sygnalizację między neuronami oraz powodując reakcję immunologiczną, prowadzącą do rozwoju stanu zapalnego, który z kolei może skutkować uszkodzeniem pobliskich komórek. Odkładają się również wokół naczyń krwionośnych, osłabiając je, co zwiększa ryzyko udaru mózgu. Peptydy Aβ pozostające wewnątrz neuronów mogą dodatkowo mieć wpływ na nadmierną aktywację ścieżki sygnałowej prowadzącej do aktywacji kinaz fosforylujących białko Tau. Wiąże się to z drugim zjawiskiem charakterystycznym dla rozwoju AD - obecnością *splątków neurofibrylarnych*.

Białko Tau odgrywa bardzo ważną rolę w systemie transportu wewnątrzkomórkowego, gdyż stabilizuje ono mikrotubule [5]. Jednak po przyłączeniu do niego grupy fosforanowej (poprzez aktywność odpowiedniej kinazy białkowej) nie jest w stanie spełniać swojej funkcji, gdyż odłącza się od mikrotubul, tym samym przestając je wspierać [6]. Ufosforylowane białka Tau grupują się razem, tworząc tzw. splątki neurofibrylarne. System transportu w komórce bez wspierających białek Tau ulega poważnym zaburzeniom, co z czasem prowadzi do apoptozy - programowanej śmierci komórki.



Rysunek 1.2: Główne geny mające związek z rozwojem choroby Alzheimera (wykres na podstawie [7]). Pozycja na wykresie zależy od powszechności wariantów danego genu sprzyjających chorobie (oś X) oraz od ryzyka zachorowania związanego z danym loci (oś Y).

Przeprowadzono wiele badań oraz analiz porównawczych genotypu osób chorych i zdrowych [8]-[9]. Dzięki nim wyróżniono grupę genów, które mogą być odpowiedzialne za patogenezę prowadzącą do rozwoju choroby Alzheimera. Dotychczas uzyskane wyniki wskazują, że w badaniach nad genetycznymi podstawami AD należy uwzględnić występowanie rodzinnej postaci tej choroby, dającej zwykle pierwsze objawy przed 65 rokiem życia (fEOAD, *familial Early-onset Alzheimer's disease*) oraz postaci sporadycznej, dającej pierwsze objawy w późniejszym

wieku (sLOAD, *sporadic Late-onset Alzheimer's disease*). fEOAD jest najczęściej chorobą autosomalną dominującą, wywoływaną mutacjami w genach APP, PSEN1 i PSEN2. Gen APP koduje wspomniane wcześniej białko APP - prekursor peptydu Aβ. PSEN1 oraz PSEN2 kodują podstawowe komponenty tworzące γ-sekretazę. Opisane do tej pory mutacje w tych genach pozwalają jednak wytłumaczyć jedynie 2-10% przypadków AD [10].

sLOAD ma o wiele bardziej złożone podłoże genetyczne. Na podstawie badań bliźniąt oraz osób spokrewnionych ustalono, że odziedziczone warianty genetyczne stanowią około 50-80% czynnika ryzyka zachorowania na LOAD [11]. Pozostała część to wpływ środowiska, taki jak dieta czy przebyte choroby. Z tego powodu jednoznaczne ustalenie przyczyn LOAD jest wyjątkowo trudne. Do tej pory znanych jest kilkadziesiąt różnych loci (Rys. 1.2), o których uważa się, że mają mniejszy bądź większy wpływ na ryzyko wystąpienia choroby. Można podzielić je na kilka grup, zależnie od procesów z jakimi są związane [12]. Pierwsza z nich to te, związane z metabolizmem cholesterolu. Wśród nich najważniejszy jest gen APOE (kodujący apolipoproteinę E), który zależnie od wariantu może zwiększać lub zmniejszać ryzyko zachorowania. ApoE jest regulatorem metabolizmu lipoprotein, mającym duży wpływ na transport cholesterolu (również w komórkach nerwowych). Pokazano również, że odgrywa rolę w regulacji procesów zapalnych [13]. Cząsteczki kodowane przez to loci wiążą się do peptydów Aβ, przez co wpływają na poziom ich agregacji w przestrzeni międzykomórkowej. Druga grupa to loci związane ze wspomnianą już odpowiedzią immunologiczną organizmu, która ma duży wpływ na rozwój AD [14]-[15]. Przykładem genu z tej grupy jest gen CR1, kodujący białko o tej samej nazwie. Białko CR1 należy do tak zwanego układu dopełniacza (*complement system*). Działanie tego układu polega na zapoczątkowaniu szeregu reakcji, prowadzących do aktywacji mechanizmów odpowiedzi immunologicznej i reakcji zapalnej. Mutacje w obrębie genu CR1 mogą zaburzać ścieżkę sygnałową wspomnianego układu wpływając tym samym na nieprawidłową odpowiedź układu odpornościowego [16]. Trzecią możliwą do wyodrębnienia grupą są loci związane z procesem endocytozy, który jest krytyczny dla poprawnego metabolizmu białka APP [15]. Przykładowe geny związane z tym procesem, a które zostały wyróżnione jako istotne w badaniach GWAS to SORL1, BIN1, PICALM i inne [8].

1.2. Poszukiwanie chorobotwórczych polimorfizmów DNA

W genetyce termin *polimorfizm* oznacza miejsce w genomie różniące się pomiędzy osobnikami danej populacji. W odróżnieniu od mutacji somatycznych, polimorfizmy są dziedziczone między osobnikami. Ponadto polimorfizmy mogą dotyczyć pojedynczego nukleotydu lub większych obszarów genomu (na przykład liczby tak zwanych powtórzeń tandemowych).

Polimorfizm pojedynczego nukleotydu (w skrócie SNP - *Single Nucleotide Polymorphism*) odnosi się, jak sama nazwa wskazuje, do miejsc w genomie, różniących się od sekwencji referencyjnej pojedynczym nukleotydem. Sekwencja referencyjna to umowny genom bazowy, względem którego opisuje się sekwencję DNA badanego osobnika. Jeśli dany polimorfizm u badanego osobnika ma taką formę, jaka występuje w genomie referencyjnym, to mówimy, że osobnik posiada *allel referencyjny*. W przeciwnym wypadku osobnik posiada *allel alternatywny*. Ponadto przyjęło się, że dany SNP musi występować u przynajmniej 1% osobników danej populacji. Jeśli jest rzadszy, dane miejsce uznaje się za mutację.

Polimorfizmy mogą występować zarówno w obrębie sekwencji kodującej białko, jak i niekodującej. Polimorfizm w sekwencji kodującej może być przyczyną różnic w budowie i funkcjonowaniu

białka, natomiast polimorfizm w obszarze niekodującym może wpływać na funkcjonowanie elementów regulatorowych.

Ze względu na stosunkowo łatwą analizę, najczęstszym przedmiotem badań, których celem jest identyfikacja genetycznego podłoża chorób, są warianty genetyczne o długości jednego nukleotydu - SNP-y lub mutacje somatyczne. Są to zwykle badania asocjacyjne całego genomu, tak zwane GWAS (*Genome Wide Association Studies*). Polegają one na porównaniu genotypów dwóch dużych grup osobników: docelowej i kontrolnej, które służy do identyfikacji wariantów, których częstość występowania istotnie różni się między tymi grupami. Genotyp badanych osobników wyznacza się najczęściej analizując jedynie ograniczony zestaw najczęściej występujących polimorfizmów (zwykle około 1 miliona). Jest to podejście skutecznie ograniczające koszty przeprowadzenia badania, ale jednocześnie związane z utratą znacznej ilości informacji. Sekwencjonowanie całogenomowe (w skrócie WGS - *Whole Genome Sequencing*) to ogólniejsze podejście, polegające na sekwencjonowaniu całych genomów pacjentów, a następnie ustaleniu różnic pomiędzy nimi a genomem referencyjnym. Oczywiście koszt takiego badania jest wielokrotnie wyższy. Jednak jeśli celem analiz jest poznanie nowych istotnych loci, bądź wcześniejsze badania typu GWAS nie przyniosły oczekiwanych wyników, to sekwencjonowanie całogenomowe może być źródłem wartościowych informacji.

Warto wreszcie zaznaczyć, że współwystępowanie danego SNP-a i choroby nie musi oznaczać, że właśnie on jest chorobotwórczy. Szczególnie w przypadku chorób o złożonym podłożu genetycznym wskazuje ono przede wszystkim, że dany rejon genomu prawdopodobnie ma wpływ na rozwój choroby (w odróżnieniu od chorób jednogenowych, gdzie pojedyncza mutacja jest ich bezpośrednią przyczyną). Ze względu na funkcjonalne powiązanie blisko położonych miejsc w genomie oraz występowanie zjawiska sprzężenia genetycznego należy się spodziewać, że SNP-y współwystępujące z chorobą będą się grupować, wskazując okolicę istotnych genów bądź obszarów regulatorowych.

1.3. Konsorcjum ADNI

Skrót ADNI pochodzi od *Alzheimer's Disease Neuroimaging Initiative* (`adni.loni.usc.edu`). Pod tą nazwą kryje się międzynarodowa prywatno-publiczna inicjatywa zapoczątkowana w 2004 roku w Stanach Zjednoczonych. Jej celem jest rozwój klinicznych, genetycznych oraz biochemicznych markerów służących do wczesnego wykrywania i śledzenia rozwoju choroby Alzheimera. W tym celu gromadzone są różnorodne dane na temat pacjentów, między innymi dane kliniczne, genetyczne, wyniki badań (zdjęcia z MRI oraz PET), próbki biologiczne. W ramach projektu do tej pory zebrano i udostępniono dane dla ponad tysiąca pacjentów, w tym dane z sekwencjonowania całogenomowego dla 808 osób. Jednym z ważniejszych zadań konsorcjum jest pośredniczenie w udostępnianiu danych pomiędzy ośrodkami badawczymi. Dzięki temu aspektowi działalności ADNI możliwe było uzyskanie danych genetycznych wykorzystanych do analiz przedstawionych w tej pracy.

Rozdział 2

Cel pracy

Celem niniejszej pracy jest stworzenie klasyfikatora pozwalającego na odróżnienie osób zdrowych od chorych na chorobę Alzheimera na podstawie danych genetycznych i wskazanie polimorfizmów genetycznych najistotniejszych z punktu widzenia procesu klasyfikacji.

Rozdział 3

Metody

Dane analizowane w tej pracy składają się z obiektów (nazywanych też próbkami), które odpowiadają pacjentom. Każdy z obiektów ma atrybuty (nazywane też predyktorami, cechami lub zmiennymi), które odpowiadają SNP-om. Dane najłatwiej jest przedstawić w postaci macierzy, której wiersze to kolejne obiekty, a kolumny to kolejne atrybuty przypisane do danych obiektów.

3.1. Metody oceny istotności atrybutów i ich selekcji

Stworzenie podzbioru najważniejszych atrybutów jest niezbędne do przeprowadzenia analizy danych, w których liczba atrybutów jest istotnie większa od liczby obiektów (sytuacja „małe n, duże p"). Zastosowanie metod klasyfikacji do tego typu zbioru danych wiąże się z ryzykiem przetrenowania modelu. Oznacza to, że klasyfikator dopasuje się ściśle do danych użytych do jego budowy, przez co nie będzie uniwersalny. Aby temu zapobiec można wykorzystać fakt, że przy bardzo dużym zbiorze atrybutów zazwyczaj nie wszystkie z nich mają wpływ na badane zjawisko. Z tego powodu warto dokonać wyboru najistotniejszych atrybutów (selekcji cech), w celu uproszczenia modelu oraz zmniejszenia ryzyka jego przetrenowania.

Selekcja cech może polegać na wyznaczeniu jak najmniejszego zbioru atrybutów, dającego jednocześnie najlepszą możliwą jakość klasyfikacji (*minimal-optimal problem*). Tego typu podejście świetnie się sprawdza, jeśli celem jest tylko i wyłącznie stworzenie dobrego klasyfikatora. Jednak w ten sposób nie da się zidentyfikować wszystkich atrybutów, które są w jakikolwiek sposób powiązane z badanym zjawiskiem. A często właśnie to jest najważniejsze w prowadzonej analizie, gdyż pozwala zrozumieć badany proces. Wtedy odpowiedniejszą strategią będzie poszukiwanie wszystkich atrybutów od których zależy badana cecha (*all-relevant*). Jest to złożone zadanie, ponieważ podczas decydowania o istotności danego atrybutu nie można opierać się tylko na zmianach w dokładności klasyfikacji stworzonej bez niego i z nim. O ile spadek dokładności po usunięciu danego atrybutu jednoznacznie wskazuje na jego powiązanie z cechą, o tyle brak tego spadku nie jest wystarczającą przesłanką do stwierdzenia jego nieistotności. Z tego powodu potrzebna jest inna miara istotności atrybutu oraz warunek mówiący dla jakiej jej wartości uznajemy dany atrybut za istotny, a kiedy jest to tylko wynik losowych fluktuacji. Metody oceny istotności atrybutów, takie jak *Gini impurity* czy *Gini Gain*, są podstawowym narzędziem służącym do wyboru podzbioru najważniejszych z nich i pozwalają zarówno na uzyskanie informacji o istotności poszczególnych atrybutów, jak i na porównywanie ich między sobą.

3.1.1. Gini impurity, Gini Gain

Gini impurity jest parametrem określającym nieczystość zbioru danych, składającego się z obiektów przypisanych do różnych klas. Wylicza się go ze wzoru:

$$GI = \sum_{k=1}^{|K|} p_k(1 - p_k)$$

Gdzie:

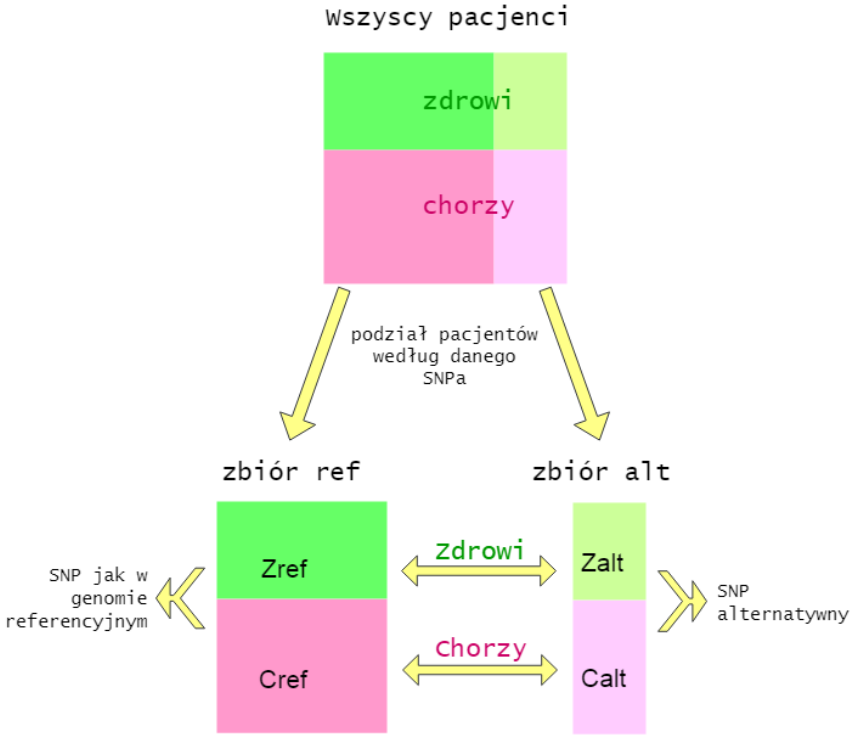
K	-	zbiór klas do których należą obiekty ze zbioru danych
p_k	-	prawdopodobieństwo wylosowania obiektu z k-tej klasy
$(1 - p_k)$	-	prawdopodobieństwo wylosowania obiektu z innej klasy niż k-ta

W szczególności jeśli do zbioru K należą tylko dwie klasy: $K = \{0, 1\}$ (jak to jest w przypadku analizy będącej tematem tej pracy):

$$GI = \sum_{k=0,1} p_k(1 - p_k) = p_0(1 - p_0) + p_1(1 - p_1) = \underbrace{p_0 + p_1}_{p_0+p_1=1} - p_0^2 - p_1^2 = 1 - p_0^2 - p_1^2$$

Gini impurity przyjmuje wartości z przedziału $[0, {}^1/_2]$, gdzie 0 oznacza zbiór czysty (składający się z obiektów należących do tej samej klasy), a ${}^1/_2$ oznacza zbiór maksymalnie nieczysty (w przypadku $K = \{0, 1\}$ jest to zbiór, w którym liczba obiektów z klasy '0' równa się liczbie obiektów z klasy '1').

Gini Gain to różnica pomiędzy końcową i początkową wartością *Gini impurity* po podziale według danego atrybutu. Jeśli wejściowy zbiór danych do klasyfikacji składa się z obiektów przypisanych do klas, to można wyliczyć dla niego GI. Po podziale tego zbioru według pewnego atrybutu, powstaje kilka podzbiorów (tyle, ile różnych wartości przyjmuje dany atrybut). Dla każdego z nich również możemy wyliczyć GI. Aby podsumować nieczystości wszystkich podzbiorów jedną wartością wykorzystuje się średnią ważoną (wagi to stosunek wielkości danego podzbioru do całości). Obliczona wartość to końcowe GI. Szukane *Gini Gain* to różnica tej wartości i początkowego GI.



Rysunek 3.1: Schemat podziału zbioru danych wejściowych na odpowiednie podzbiory według wybranego SNP-a.

W przypadku danych analizowanych w tej pracy, kolejne kroki obliczania *Gini Gain* dla danego SNP-a wyglądają następująco (wyjaśnienie oznaczeń - Rys. 3.1):

1. Wartość GI wejściowego zbioru danych:

$$GI(wszyscy) = 1 - \left(\frac{|zdrowi|}{|wszyscy|}\right)^2 - \left(\frac{|chorzy|}{|wszyscy|}\right)^2$$

2. Wartości GI podzbiorów powstałych po podziale według danego SNP-a:

$$GI(ref) = 1 - \left(\frac{|Zref|}{|ref|}\right)^2 - \left(\frac{|Cref|}{|ref|}\right)^2$$

$$GI(alt) = 1 - \left(\frac{|Zalt|}{|alt|}\right)^2 - \left(\frac{|Calt|}{|alt|}\right)^2$$

3. *Gini Gain* to różnica pomiędzy GI dla wejściowego zbioru danych a średnią ważoną GI uzyskanych podzbiorów:

$$GG(\text{SNP}) = GI(wszyscy) - \Big(\underbrace{\frac{|ref|}{|wszyscy|}}_{\text{waga zbioru } ref} GI(ref) + \underbrace{\frac{|alt|}{|wszyscy|}}_{\text{waga zbioru } alt} GI(alt) \Big)$$

Warto zauważyć, że GG jest zawsze ≥ 0.

Różne SNP-y mają różne wartości *Gini Gain* - *GG* jest tym większe, im większy wpływ ma dany SNP na czystość zbiorów po podziale, czyli na dokładność klasyfikacji. W ten sposób łatwo jest porównywać atrybuty między sobą.

3.1.2. P-wartość

P-wartość, nazywana też *prawdopodobieństwem testowym*, to prawdopodobieństwo, że wartości zaobserwowane dla danego atrybutu wystąpiły przypadkowo wskutek losowej zmienności prób i są całkowicie niezależne od badanej cechy. Odnosząc to do danych analizowanych w tej pracy: *p-wartość* danego SNP-a mówi o tym, jakie jest prawdopodobieństwo, że wartości przyjmowane przez ten SNP (referencyjne/alternatywne) u przebadanych pacjentów są zupełnie przypadkowe i niezależne od tego, czy konkretny pacjent jest chory czy zdrowy. Czyli im mniejsze jest to prawdopodobieństwo, tym pewniejsze jest, że dany SNP jest istotny w procesie rozwoju choroby Alzheimera. *P-wartość* wylicza się na podstawie testu statystycznego, na przykład na podstawie jednostronnego testu χ^2.

3.1.3. Algorytm Boruta

Algorytm Boruta, stworzony przez M.Kursę oraz W.Rudnickiego [17], jest metodą wyboru najistotniejszych predyktorów, opartą na klasyfikatorze lasów losowych. Do oceny atrybutów wykorzystuje parametr wyliczany na podstawie *ważności*, która jest miarą istotności atrybutów w lasach losowych (zobacz 3.2.2), będąca średnią wartości *Gini Gain* (zobacz 3.1.1) dla danego atrybutu. Wadą *ważności* jest to, że jej zastosowanie wiąże się z utratą informacji na temat wahań istotności danego atrybutu pomiędzy drzewami w danym lesie losowym. Z tego powodu algorytm Boruta wykorzystuje parametr będący jej modyfikacją - *Z-score* - zdefiniowany jako *ważność* podzielona przez odchylenie standardowe wartości istotności danego atrybutu (*Gini Gain*) pomiędzy drzewami:

$$Z = \frac{\overline{GG}}{SD(\mathrm{GG})}$$

Gdzie:

- $\overline{GG}$ - średnia z wartości *Gini Gain* wyliczonych na podstawie każdego wystąpienia danego atrybutu w lesie
- $SD(\mathrm{GG})$ - odchylenie standardowe zbioru wartości *Gini Gain* dla danego atrybutu

Jednak również sam *Z-score* wyliczony na podstawie jednego lasu losowego nie jest wystarczający do podjęcia decyzji o odrzuceniu bądź zachowaniu atrybutu. Wciąż może się zdarzyć sytuacja, w której atrybut niezwiązany z diagnozą został losowo przyporządkowany do takich drzew w lesie, że jego *Z-score* jest wysoki. Jednocześnie każdy, nawet losowy atrybut osiągnie *Z-score* ≥ 0 (wynika to z własności wartości *Gini Gain* - zobacz 3.1.1). Z tego powodu pojawia się pytanie o minimalne *Z-score*, od którego można uznać dany atrybut za rzeczywiście związany z badaną cechą. Potrzebna jest zewnętrzna miara pozwalająca ocenić, czy otrzymany *Z-score* jest lepszy, niż wartość wyliczona na podstawie losowych danych. Żeby ją wyznaczyć dla każdego atrybutu tworzony jest dodatkowy, odpowiadający mu atrybut zwany *cieniem*, którego wartości są otrzymane poprzez przetasowanie wartości oryginalnego atrybutu między obiektami, a więc są losowe. Następnie na wszystkich atrybutach (oryginalnych i losowych) budowany jest klasyfikator oraz obliczane są ich *Z-score*. *Ważność* któregokolwiek z cieni może być niezerowa tylko na skutek losowych fluktuacji. Dzięki temu zbiór losowych atrybutów

wykorzystuje się jako punkt odniesienia przy podejmowaniu decyzji, czy *Z-score* oryginalnych atrybutów są znaczące, czy też wynikają z losowości danych. Algorytm Boruta składa się z wielu iteracji, więc prawdopodobieństwo, że jakiś oryginalny atrybut niezwiązany z diagnozą będzie w wielu z nich lepszy od wszystkich cieni, jest bardzo małe.

Algorytm Boruta składa się z następujących kroków:

1 Dodanie do zbioru danych *cieni* stworzonych na podstawie oryginalnych atrybutów.

2 Budowa klasyfikatora lasów losowych, obliczenie *Z-score* dla każdego atrybutu.

3 Znalezienie najwyższego *Z-score* spośród *cieni* i oznaczenie go jako MZSA.

4 Przypisanie tak zwanego *hit* do każdego atrybutu, którego *Z-score* jest wyższy od MZSA.

5 W celu uznania atrybutu za istotny: wyliczenie dla niego *p-wartości* na podstawie testu dwumianowego, gdzie:

(a) liczba sukcesów k to liczba *hit-ów* uzyskanych do tej pory przez rozważany atrybut

(b) liczba prób n to liczba wykonanych dotychczas iteracji

(c) prawdopodobieństwo sukcesu p jest równe $^1/_2$

(d) hipoteza alternatywna to $p > \frac{1}{2}$

6 Akceptacja atrybutów, których *p-wartość* jest mniejsza od 0.01.

7 W celu uznania atrybutu za nieistotny: wyliczenie dla niego *p-wartości* na podstawie testu podobnego do opisanego w punkcie **5**, zmieniając jedynie hipotezę alternatywną na $p < \frac{1}{2}$.

8 Odrzucenie atrybutów, których *p-wartość* jest mniejsza od 0.01.

9 Usunięcie ze zbioru danych atrybutów zaakceptowanych, odrzuconych oraz *cieni*.

10 Określenie pozostałych atrybutów jako niepewnych - na nich opiera się kolejna iteracja algorytmu.

11 Powtarzanie algorytmu aż do osiągnięcia kryterium zatrzymania (którym jest określenie istotności wszystkich atrybutów bądź wykonanie maksymalnej liczby iteracji).

W praktyce powyższy algorytm poprzedzają trzy nieco zmienione iteracje. W tych iteracjach atrybuty porównywane są z piątym, trzecim i drugim najlepszym *cieniem* - dzięki temu warunki przypisywania atrybutom *hit-ów* są mniej restrykcyjne. Test stosowany w celu odrzucenia atrybutów (punkt **7**) wykonuje się na końcu każdej z iteracji wstępnych, natomiast test wykonywany w celu akceptacji (punkt **5**) nie jest przeprowadzany. Dzięki takim wstępnym powtórzeniom zwiększa się liczba prób n, dzięki czemu zmniejsza się wariancja wyniku testu akceptującego atrybuty w kolejnych iteracjach.

3.2. Budowa klasyfikatora

Następnym krokiem po wyborze najistotniejszych atrybutów jest zbudowanie na ich podstawie odpowiedniego klasyfikatora. W tym celu w niniejszej pracy wykorzystane zostały metody

drzewiaste: pojedyncze drzewo decyzyjne oraz las losowy. Są one proste do zrozumienia i interpretacji oraz łatwo jest w ich przypadku uzyskać informację dotyczącą tego, na jakiej podstawie podjęta została decyzja o klasyfikacji. Są to cechy bardzo istotne, jeśli tworzenie klasyfikatora ma się przyczynić do lepszego zrozumienia analizowanego zjawiska. Ponadto działają szybko w porównaniu do innych metod klasyfikacji, takich jak liniowa analiza dyskryminacyjna (znana jako LDA) bądź sieci neuronowe, co również jest nie bez znaczenia przy analizie danych charakteryzujących się bardzo dużą liczbą atrybutów.

3.2.1. Drzewa decyzyjne

Najprostszą z metod drzewiastych jest budowa pojedynczego drzewa klasyfikacji. Polega ona na znalezieniu takich podziałów badanego zbioru danych na podzbiory, który maksymalizuje dokładność klasyfikacji. W tym przypadku klasyfikacja polega na przypisaniu do wszystkich obiektów z danego podzbioru tej samej, najczęściej w nim występującej klasy.

Proces budowy drzewa decyzyjnego można przedstawić jako rekurencyjny algorytm podziału zbioru danych na podzbiory. Każdy podział wykonuje się według wartości jednego z atrybutów. W celu wyboru najlepszego możliwego podziału potrzebna jest miara jego jakości, którą wyznacza się na podstawie zmiany czystości zbioru przed i po podziale. W drzewach klasyfikacji najczęściej wykorzystuje się następujące miary czystości zbioru:

- *Gini impurity* - omawiana wcześniej miara nieczystości zbioru (zobacz 3.1.1).
- *Entropia krzyżowa* - miara nieuporządkowania układu. Im mniejsza, tym zbiór jest bardziej uporządkowany (zawiera więcej elementów z tej samej klasy). *Entropię krzyżową* oblicza się ze wzoru:

$$D = -\sum_{k=1}^{|K|} p_k \log p_k$$

Gdzie:

K	-	zbiór klas do których należą obiekty ze zbioru danych
p_k	-	prawdopodobieństwo wylosowania obiektu z k-tej klasy

Podsumowując, proces budowy drzewa dla danych składających się z elementów, których atrybuty są oznaczone poprzez: $X_1, X_2, ..., X_p$, można przedstawić rekurencyjnie:

1 Dla każdego atrybutu x, gdzie $x \in \{X_1, X_2, ..., X_p\}$:

(a) Dla każdej wartości $v \in V$, gdzie zbiór V jest zbiorem wartości, które przyjmuje atrybut x:

i. Podziel zbiór obiektów według wartości v atrybutu x.

ii. Policz jakość podziału - różnicę pomiędzy czystością zbioru wejściowego i powstałych podzbiorów (*Gini impurity* bądź *entropia krzyżowa*).

2 Podziel obiekty na dwa podzbiory według atrybutu o najwyższej jakości podziału.

3 Powtarzaj kroki **1-2** aż do osiągnięcia kryterium zatrzymania.

Kryterium zatrzymania może być na przykład minimalna liczba elementów w pojedynczym podzbiorze albo maksymalna głębokość drzewa.

Po ustaleniu kolejności i rodzajów podziałów (w przypadku atrybutów o niebinarnych wartościach - rodzaj podziału oznacza tu wartość danego atrybutu, według której dzielimy obiekty) otrzymujemy gotowe drzewo, które łatwo da się przedstawić w postaci graficznej. Proces klasyfikacji na jego podstawie polega na „odszukaniu" do którego podzbioru należy rozważany obiekt.

3.2.2. Algorytm lasów losowych

Lasy losowe są rozwinięciem metody drzew decyzyjnych. Wadą drzew jest ich duża wariancja. Oznacza to, że jeśli podzielimy dane wejściowe na dwa zbiory i dla każdego z nich zbudujemy drzewo, jest duża szansa, że wyniki będą całkiem inne niż dla pojedynczego drzewa zbudowanego na podstawie całych danych. Nie świadczy to dobrze o stworzonym modelu, a tym bardziej o poprawności wyciągniętych na jego podstawie wniosków na temat zależności pomiędzy atrybutami a badaną cechą. Algorytm lasów losowych radzi sobie z tym problemem w ten sposób, że buduje wiele drzew (stąd nazwa „las"), z których każde bazuje na losowym podzbiorze atrybutów (stąd „losowy") o określonej wielkości. Ostateczna decyzja gdzie przypisać analizowany obiekt zależy od tego, do której klasy był przypisywany najczęściej.

Podsumowując, proces budowy lasu losowego na podstawie danych składających się z elementów, których atrybuty są oznaczone poprzez: $X_1, X_2, ..., X_p$, można przedstawić następująco:

Powtarzaj n razy:

1. Wylosuj m atrybutów ze zbioru $\{X_1, X_2, ..., X_p\}$.
2. Na podstawie wybranych atrybutów zbuduj drzewo (proces budowy drzewa - zobacz 3.2.1).
3. Dodaj stworzone drzewo do zbioru drzew składających się na tworzony las losowy.

Gdzie:

- n - liczba drzew w lesie, zazwyczaj przyjmuje się $n = 10$
- m - liczba atrybutów losowanych do budowy pojedynczego drzewa, zazwyczaj $m = \sqrt{p}$, gdzie p to liczba wszystkich atrybutów

W celu przypisania danego elementu do klasy las losowy przeprowadza poniższą procedurę:

Dla każdego drzewa w lesie:

1. Przekaż do drzewa analizowany element w celu przypisania go do klasy.
2. Zapisz uzyskaną klasę.

Przypisz element do najczęściej wybieranej przez drzewa klasy.

Ważność atrybutów w lesie losowym

Parametr *ważność*, przypisany do każdego atrybutu w lesie losowym, mówi o średnim spadku dokładności klasyfikacji, po zastąpieniu tego atrybutu losowymi danymi. Dzięki temu niesie informacje o tym, jak dużą rolę odgrywa dany atrybut w stworzonym klasyfikatorze. Wylicza

się go na podstawie miary *Gini Gain* (zobacz 3.1.1). Jednak w lesie dany atrybut występuje wiele razy (w różnych drzewach), więc uzyskujemy wiele wartości *GG*. Szukaną *ważnością* jest ich średnia, którą interpretuje się jako średni wzrost czystości zbioru po podziale według danego atrybutu.

Wartość *ważności* rozumianej w ten sposób nie pokrywa się z wartością *Gini Gain* wyliczoną na podstawie całych danych. Różnica tkwi w analizowanych zbiorach, na których podstawie wyliczamy *GG*. W lesie losowym liczymy *GG* oparte na różnych podzbiorach zbioru wejściowego. Skład tych podzbiorów zależy od „kontekstu", czyli wcześniejszych podziałów zbioru danych, według innych atrybutów rozważanych w tym drzewie. Tylko pierwszy podział w drzewie jest wykonywany na całych danych. Reszta podziałów zawsze jest rozważana w jakimś kontekście. Co więcej ten kontekst zależy od **losowego** wyboru grup atrybutów na podstawie których budowane są drzewa. Z tego powodu wariancja wartości *ważności* jest tym większa, im więcej jest rozważanych atrybutów oraz im mniej jest drzew w lesie. Im więcej atrybutów, tym większa losowość przy ich grupowaniu, a co za tym idzie większe fluktuacje *Gini Gain* oraz większa wariancja *ważności*. Im mniej drzew tym mniej szans na wystąpienie danego atrybutu, co wiąże się z mniejszą liczbą wartości *Gini Gain* wykorzystanych do wyliczenia *ważności*.

3.3. Ocena jakości klasyfikacji

Ostatnim krokiem składającym się na budowę modelu jest ocena jakości klasyfikacji stworzonego narzędzia, czyli kontrola jak radzi on sobie w praktyce. Niezbędne do tego jest rozróżnienie pomiędzy danymi treningowymi a testowymi. Dane treningowe to te, które wykorzystuje się do budowy modelu. W przypadku metod drzewiastych są to dane, na podstawie których podejmuje się decyzje na temat podziałów zbioru elementów. Dane testowe są przeznaczone tylko do testowania modelu i nie mogą być brane pod uwagę podczas trenowania klasyfikatora. Dzięki temu ocena jakości klasyfikatora na ich podstawie jest wiarygodna. Dane testowe niejako „symulują" prawdziwe dane, z tą różnicą, że dla elementów ze zbioru testowego wiemy dokładnie do jakiej klasy należą.

3.3.1. Wynik testowy i treningowy

Wynik testowy mówi o tym, jaka część danych testowych została poprawnie sklasyfikowana. Mając gotowy model, klasyfikuje się na jego podstawie elementy ze zbioru testowego. Następnie porównuje się uzyskane klasy z tymi, do których naprawdę one należą. Wynik testowy to procent elementów przypisanych właściwie.

Z kolei wynik treningowy mówi o tym, jaka część danych treningowych została poprawnie sklasyfikowana.

3.3.2. Walidacja krzyżowa

Często wynik testowy nie jest wystarczający do rzetelnej oceny modelu, gdyż cechuje się wysoką wariancją. Ma to związek z tym, że zbiór danych testowych jest ograniczony. Im więcej danych przeznaczymy do testowania, tym mniej będzie danych treningowych, od których zależy dokładność klasyfikatora. Rozwiązaniem tego problemu jest *walidacja krzyżowa*. Metoda ta polega na podziale zbioru danych na ustaloną liczbę równych części. Następnie każdy z podzbiorów staje się zbiorem testowym dla klasyfikatora zbudowanego na podstawie pozostałych

danych. W ten sposób otrzymuje się tyle wyników testowych, na ile podzbiorów dane zostały podzielone. Z tego łatwo wyliczyć średnią i w ten sposób uzyskać wynik testowy oparty na znacznie większej liczbie prób.

Estymacja wyniku testowego na podstawie *walidacji krzyżowej* przebiega następująco:

1 Podział zbioru danych na n równych części.

2 Wykonanie dla każdej i-tej części ($i = 1, 2, ..., n$) następujących kroków:

(a) Zbudowanie klasyfikatora na podstawie wejściowego zbioru danych pomniejszonego o i-tą część.

(b) Policzenie wyniku testowego stworzonego modelu wykorzystując jako zbiór testowy i-tą część danych.

3 Zwrócenie jako wyniku średniej z uzyskanych wyników testowych:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} T_i$$

Gdzie:

n	-	mówi o krotności walidacji (z ilu prób szacowany jest wynik testowy)
T_i	-	wynik testowy dla i-tego zbioru testowego

Rozdział 4

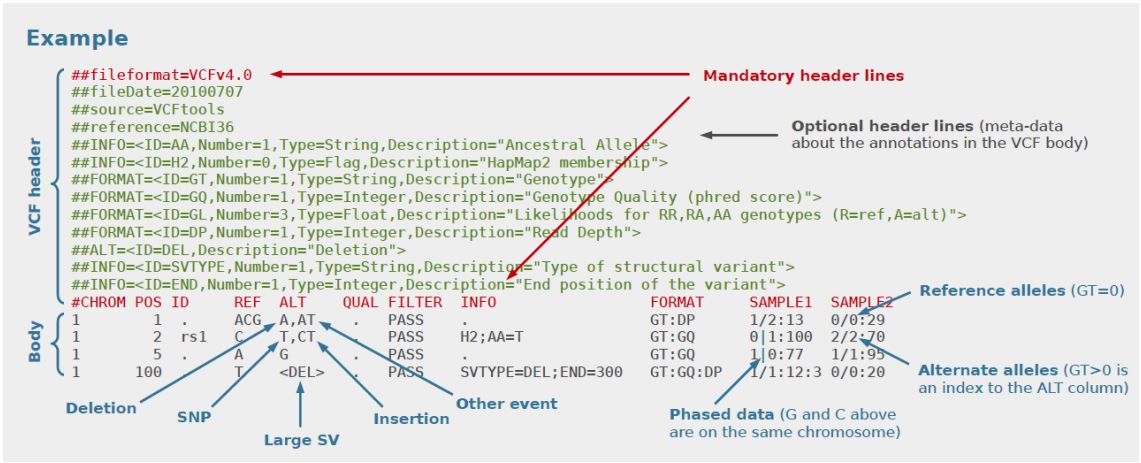
Implementacja

4.1. Przygotowanie danych z ADNI

4.1.1. Analiza plików `vcf`

Konsorcjum ADNI udostępnia swoje dane w postaci spakowanych do archiwum plików `vcf`. Skrót `vcf` pochodzi od *Variant Call Format*. Jest to typ danych służący do zapisu wariantów sekwencji DNA (Rys. 4.1). Dzięki specyficznej strukturze pozwala przechować dużo informacji w stosunkowo małej pamięci.

W udostępnionych plikach są zapisane różnice (takie jak SNP-y, insercje, delecje) w genomach pacjentów w porównaniu do genomu referencyjnego GRCh37 (hg19), opracowanego przez *Genome Reference Consortium*. Jeden plik dotyczy jednego chromosomu.



Rysunek 4.1: Przykładowy fragment pliku `vcf`. W przypadku plików z ADNI kolejne wiersze w sekcji *Body* dotyczą kolejnych fragmentów genomu innych od genomu referencyjnego. Próbek (kolumny o nazwach *Sample*) jest tyle, ilu było badanych pacjentów. Źródło: [18]

Pierwszym krokiem do uzyskania danych potrzebnych do wykonania opisanej w tej pracy analizy, było wybranie z pobranych plików tylko różnic jednonukleotydowych, czyli SNP-ów. W tym celu wykorzystano metodę `Select Variants` z pakietu narzędzi do analizy danych z sekwencjonowania nowej generacji - *The Genome Analysis Toolkit* [19]. W ten

sposób dla każdego chromosomu uzyskano plik `vcf` zawierający tylko warianty polimorfizmów jednego nukleotydu. Liczbę wariantów zlokalizowanych na poszczególnych chromosomach przedstawiono w tabeli 4.1.

Tabela 4.1: Liczba SNP-ów zlokalizowanych na poszczególnych chromosomach

chromosom	**liczba SNP-ów**
1	2 931 252
2	3 193 926
3	2 665 753
4	2 629 372
5	2 412 582
6	2 316 014
7	2 159 228
8	2 106 252
9	1 628 235
10	1 809 029
11	1 842 462
12	1 752 172
13	1 315 109
14	1 209 284
15	1 096 882
16	1 221 488
17	1 043 997
18	1 041 106
19	852 390
20	821 218
21	504 291
22	506 555
X	1 384 972
łącznie	38 443 569

Następnie na podstawie każdego ze stworzonych plików `vcf` wygenerowano trzy tabele:

1 Macierz M, zawierającą informacje na temat występowania SNP-ów u poszczególnych badanych osób. Jej wiersze reprezentują kolejnych pacjentów, kolumny kolejne SNP-y. W każdym polu macierzy M znajdują się dwie cyfry oddzielone ukośnikiem. Odpowiadają one dwóm allelom danego SNP-a (zależnie od kolumny), występującym u danego pacjenta (zależnie od wiersza). Możliwe są następujące wartości: -1 oznaczające brak danych; 0 oznaczające występowanie allelu referencyjnego; 1, 2 lub 3 oznaczające występowanie jednego z alleli alternatywnych.

2 Tabela zawierająca dokładne pozycje poszczególnych SNP-ów w genomie oraz odpowiadające im allele referencyjne i alternatywne.

3 Lista numerów identyfikacyjnych pacjentów (w bazie ADNI są to tak zwane numery RID) w odpowiedniej kolejności, odpowiadającej kolejności zapisu w pozostałych plikach.

W ten sposób wszystkie informacje zawarte w plikach `vcf` zostały przedstawione w sposób

znacznie ułatwiający dalszą analizę. Następnie na podstawie tych danych stworzono dla każdego chromosomu macierze X i y, które stanowiły dane wejściowe dla algorytmu wyboru najlepszych atrybutów.

Macierz X jest skróconą formą wyżej opisanej macierzy M. W każdej pozycji tabeli znajduje się tylko jedna cyfra. Jeśli oba allele z rozważanej pozycji są referencyjne, to w macierzy zapisana jest cyfra 0. Jeśli tylko jeden z alleli ma wartość referencyjną, w tabeli znajduje się wartość allelu alternatywnego. W przypadku obu alleli alternatywnych wpisana jest wartość pierwszego z nich.

Macierz y zawiera diagnozy pacjentów. Składa się z jednej kolumny, wiersze odpowiadają kolejnym osobom. Zawiera wartości: 0 - jeśli pacjent jest zdrowy, 1 - w przypadku osoby chorej na AD. Do stworzenia tej macierzy wykorzystano plik z diagnozami, którego uzyskanie zostało opisane w sekcji 4.1.2 oraz przygotowaną wcześniej tabelę z numerami RID.

Poniżej przedstawiono skrypt w języku *Bash* wykorzystany podczas wykonywania powyżej opisanej procedury: linie 2-10 - wyodrębnienie z pobranych plików `vcf` tylko zmian jednonukleotydowych, linia 11 - zamiana danych z plików `vcf` na trzy tabele opisane powyżej, linia 12 - utworzenie macierzy X, linia 13 - utworzenie macierzy y.

```
1  #!/bin/bash
2  istart="ADNI.808_indiv.minGQ_21.pass.ADNI_ID.chr"
3  iend=".vcf"
4  ostart="ADNI_chr"
5  oend="_SNPsMNPs.vcf"
6  for ((i=1; $i<24; i++)); do
7    input=${istart}${i}${iend}
8    output=${ostart}${i}${oend}
9    java -jar GenomeAnalysisTK.jar -T SelectVariants -R Homo_sapiens_assembly19.
       fasta -V $input -o $output -selectType SNP
10   echo '$i SNPs.vcf done!'
11   python vcf_to_matrix.py -chr $i
12   python makeY.py -chr $i
13   python makeX.py -chr $i
14 done
```

4.1.2. Ustalenie diagnoz pacjentów

Do stworzenia klasyfikatora niezbędna jest nie tylko informacja na temat cech przypisanych do każdego obiektu, ale również na temat przypisania obiektów do klas. W przypadku danych z ADNI klasa jest równoważna diagnozie pacjenta. Ustalenie ostatecznej klasy obiektów wymagało poznania konwencji zapisu diagnozy w każdej edycji programu ADNI (do tej pory przeprowadzono trzy edycje: ADNI1, ADNIGO pokrywające się z ADNI2 oraz ADNI3). Dodatkowo w każdej edycji do danego pacjenta przypisanych jest wiele wizyt, z których każda ma osobną diagnozę. Konsorcjum udostępnia plik z podsumowaniem wszystkich wizyt wchodzących w skład każdej z edycji. Na podstawie tych informacji dla każdego pacjenta wybrano najświeższą diagnozę z najnowszej możliwej edycji (nie każda osoba brała udział we wszystkich edycjach programu). Następnym krokiem był wybór tylko tych osób, dla których przeprowadzono sekwencjonowanie całogenomowe. Łącznie jest ich 808. Wśród nich diagnozy rozkładają się następująco:

235 - choroba Alzheimera (AD)
322 - łagodne zaburzenia poznawcze (MCI)
251 - zdrowi (NL)

Do budowy klasyfikatora, którego zadaniem było odróżnienie osób zdrowych od chorych na Alzheimera wykorzystano jedynie dane pochodzące od osób z diagnozą NL lub AD (łącznie 486 pacjentów).

4.2. Wybór najistotniejszych atrybutów

Ze względu na bardzo dużą liczbę SNP-ów w genomie (około 38 mln, 76 mln atrybutów po uwzględnieniu atrybutów-cieni dodawanych przez algorytm Boruta) proces wyboru najistotniejszych atrybutów został podzielony na etapy. Dane wejściowe do każdego etapu stanowił podzbiór SNP-ów o stałej liczebności. Liczebność tego podzbioru jest ważnym parametrem, ponieważ im więcej wejściowych atrybutów, tym więcej odpowiadających im cieni. Z kolei im więcej cieni, które z założenia są atrybutami losowymi, tym większa jest szansa, że któryś z nich osiągnie wysoki *Z-score*, a co za tym idzie będzie powodem uznania prawdziwych atrybutów za nieistotne (zobacz 3.1.3). Aby zmniejszyć wpływ losowości na decyzję o akceptacji/odrzuceniu atrybutu zbiór wejściowy powinien być możliwie mały. Z drugiej strony wynikiem działania algorytmu Boruta są najistotniejsze atrybuty w kontekście, którym jest wejściowy zbiór. Z tego powodu nie powinien być on zbyt mały. Im mniej wejściowych atrybutów, tym mniejsza szansa, że w losowym podzbiorze, na którego podstawie budowane jest drzewo „spotkają się" te skorelowane, niosące łącznie dużo informacji. Z tych powodów, po przetestowaniu algorytmu na różnych liczebnościach wybieranych podzbiorów atrybutów, wybrano 5000 jako optymalną wartość tego parametru.

Kolejnym krokiem był wybór odpowiednich parametrów Boruty. Jednym z nich jest parametr `perc`, mówiący od jakiego percentyla cieni dany atrybut musi być lepszy, aby w danej iteracji przypisano do niego *hit*. Od liczby *hit-ów* zależy szansa na uzyskanie wystarczająco niskiej *p-wartości*, a co za tym idzie akceptacji danego atrybutu (zobacz 3.1.3). Wartością domyślną jest 100. W celu uzyskania większej liczby atrybutów uznanych za istotne postanowiono przeprowadzać analizę Boruta czterokrotnie: dla `perc` $\in \{80, 90, 95, 100\}$. Cały przebieg procesu wyboru najistotniejszych atrybutów przedstawia poniższy pseudokod:

Dla każdego chromosomu **rób**:
    Dopóki nie skończą się SNP-y z danego chromosomu **rób**:
        lista <- zbiór kolejnych 5 tyś SNP-ów
        Dla p należącego do $\{80, 90, 95, 100\}$ **rób**:
            Przeanalizuj SNP-y z *listy* za pomocą Boruty z parametrem `perc`$=p$
            Zapisz wybrane SNP-y

4.3. Budowa klasyfikatora

Na podstawie czterech uzyskanych zbiorów najistotniejszych SNP-ów (wybranych przez Borutę z różnymi wartościami parametru `perc`) zbudowano cztery klasyfikatory. Do ich budowy użyte zostały funkcje *DecisionTreeClassifier* oraz *RandomForestClassifier* z biblioteki Pythona *scikit-learn*.

Parametry funkcji budowy drzewa decyzyjnego pozostały domyślne. Oznacza to, że gałęzie drzewa są dzielone do czasu, aż więzły są czyste (zawierają elementy tylko z jednej klasy), a kryterium miary jakości podziału jest *Gini Gain*. W kodzie wygląda to następująco:

```
> dt = tree.DecisionTreeClassifier()
```

W funkcji tworzącej lasy losowe został zmieniony jeden parametr - `n_estimators` - który mówi o liczbie drzew w lesie. Domyślnie jest ich 10, jednak ze względu na dużą liczbę predyktorów jego wartość ustawiono na 500. Pozostałym parametrom przypisano wartości domyślne:

```
> rf = RandomForestClassifier(n_estimators=500)
```

4.3.1. Ocena jakości klasyfikacji

Jakość klasyfikacji jest oceniania na podstawie wyniku treningowego, testowego oraz walidacji krzyżowej (zobacz sekcja 3.3). Aby wynik był miarodajny proces budowy i oceny klasyfikatora powtórzono wielokrotnie (10-krotną walidację krzyżową przeprowadzono 10 razy, wynik treningowy i testowy szacowano na podstawie 100 powtórzeń).

Poniżej zamieszczono fragment kodu w języku *Python* prezentujący funkcję *score* liczącą wynik treningowy, testowy i walidacji krzyżowej dla stworzonego klasyfikatora lasów losowych, na podstawie podanej macierzy X i y (opisanych powyżej w sekcji 4.1.1).

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score

def score(X,y,iterations=100):
    rf = RandomForestClassifier(n_estimators=500)
    scores = [ [], [], [] ]
    for i in range(iterations):
        X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=.1)
        rf.fit(X_train,y_train)
        scores[0].append(rf.score(X_train,y_train))
        scores[1].append(rf.score(X_test,y_test))
    for i in range(iterations//10):
        cc = cross_val_score(rf,X,y,cv=10)
        for el in cc:
            scores[2].append(el)
    return list(map(np.mean,scores))
```

4.3.2. Dalsza poprawa jakości klasyfikacji

Dodatkowa ocena istotności już wybranych przez Borutę atrybutów daje możliwość uzyskania dalszej poprawy jakości klasyfikacji. Dzięki usunięciu ze zbioru danych części niewiele wnoszących

atrybutów zmniejszy się przetrenowanie modelu. Do uszeregowania SNP-ów pod względem ich ważności zastosowano miarę istotności atrybutów wykorzystywaną przez lasy losowe - *Gini Gain* (zobacz sekcja 3.2.2). Metoda `RandomForestClassifier` ma tę zaletę, że automatycznie wylicza *Gini Gain* dla każdego SNP-a podczas budowy lasu. Wystarczy odwołać się do atrybutu `feature_importances_` obiektu klasyfikatora.

Mając listę SNP-ów uszeregowanych od najistotniejszych do najmniej istotnych łatwo jest zbudować klasyfikator oparty tylko na części z nich. W celu stwierdzenia jaka część najlepszych atrybutów daje najlepszy model (dobrze wytrenowany, ale nie przetrenowany) zbudowano i oszacowano dokładność klasyfikatorów zbudowanych na podstawie 20 różnej wielkości podzbiorów atrybutów ($5, 10, 15, ..., 90, 95, 100$ procent SNP-ów wybranych przez Borutę z parametrem `perc`=100). Ostatecznym klasyfikatorem został ten o najwyższym wyniku walidacji krzyżowej.

4.4. Analiza wybranych SNP-ów

Oprócz istotności danego SNP-a w kontekście pewnego podzbioru SNP-ów warto zbadać również jego istotność w kontekście całego genomu. Pozwoli to na rozróżnienie atrybutów istotnych bez względu na wartości innych SNP-ów od tych, mających duży wpływ tylko w pewnym kontekście. Do tego celu wykorzystano miary istotności predyktorów: *Gini Gain* oraz *p-wartość* (zobacz sekcję 3.1). *Gini Gain* wyliczono wprost ze wzoru, osobno dla każdego SNP-a i pełnego zbioru pacjentów (w związku z tym brak „kontekstu" podziału, który występuje w przypadku *Gini Gain* liczonego na podstawie lasów losowych - sekcja 3.2.2). *P-wartość* została ustalona dla każdego SNP-a na podstawie jednostronnego testu χ^2. Wykorzystano do tego funkcję `chisquare` z pakietu `scipy.stats`.

Rozdział 5

Wyniki

5.1. Wybór najistotniejszych atrybutów, budowa modeli

W wyniku procesu wyboru najważniejszych atrybutów wyodrębniono cztery grupy SNP-ów - każda z nich została uzyskana za pomocą metody Boruta z inną wartością parametru `perc` (80, 90, 95 i 100). Następnie na podstawie SNP-ów z każdej z grup stworzono dwa rodzaje modeli: drzewa decyzyjne oraz klasyfikatory lasów losowych (zobacz sekcję 3.2). Oszacowana dokładność uzyskanych modeli mieści się w zakresie odpowiednio 58-68% oraz 90-98% (tabela 5.1).

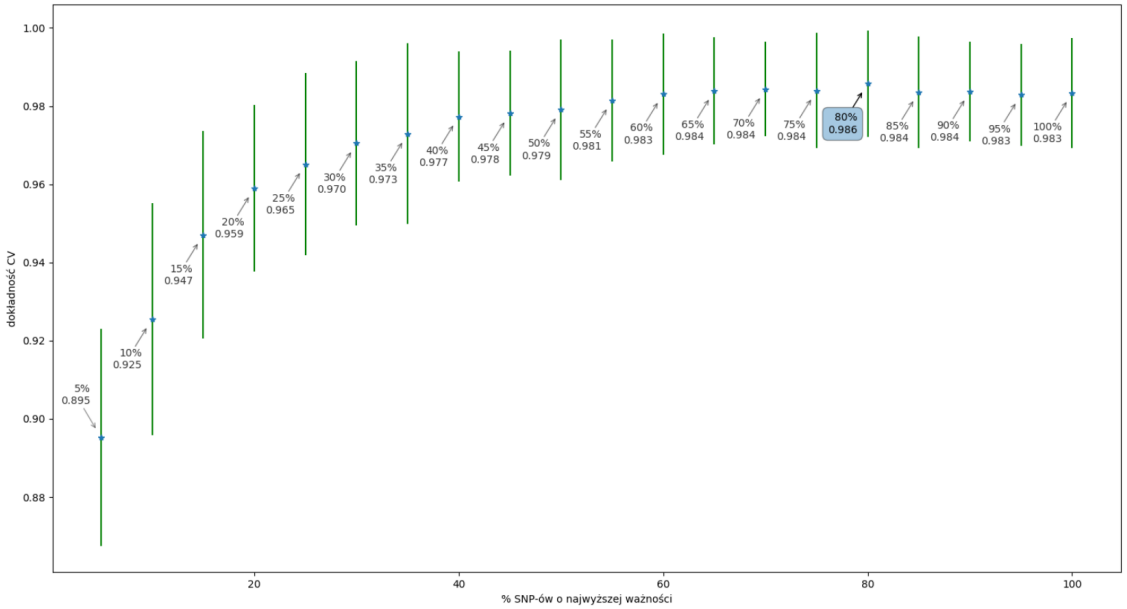
Najlepszy wynik dał model zbudowany na podstawie najmniej licznej grupy SNP-ów (uzyskanej przy `perc`=100). Zastosowanie algorytmu Boruta pozwoliło zatem zredukować liczbę atrybutów z ponad 38 mln do 7195. Tę grupę polimorfizmów wykorzystano do dalszych analiz, w tym do próby uzyskania dodatkowej poprawy jakości klasyfikacji, poprzez wybór atrybutów najbardziej istotnych z punktu widzenia klasyfikatora lasu losowego.

Tabela 5.1: W tabeli przedstawiono liczbę wybranych przez Borutę SNP-ów w zależności od parametru `perc` (kolumna **perc** i **SNP-y**) oraz dokładność klasyfikacji dwóch modeli zbudowanych na ich podstawie (kolumna **drzewo decyzyjne** i **las losowy**). Zbiór testowy stanowi 10% danych, wynik testowy (kolumny **testowy**) to wartość uśredniona po 100 powtórzeniach. Wynik CV (kolumny **CV**) został wyliczony na podstawie 10-krotnej walidacji krzyżowej, wartość uśredniona po 10 powtórzeniach.

perc	**SNP-y**	**drzewo decyzyjne**		**las losowy**	
		testowy	CV	testowy	CV
80	1 507 307	0.601	0.577	0.890	0.899
90	341 731	0.602	0.593	0.958	0.965
95	129 214	0.625	0.598	0.969	0.971
100	7 195	0.672	0.687	0.983	0.983

5.1.1. Dodatkowa selekcja atrybutów

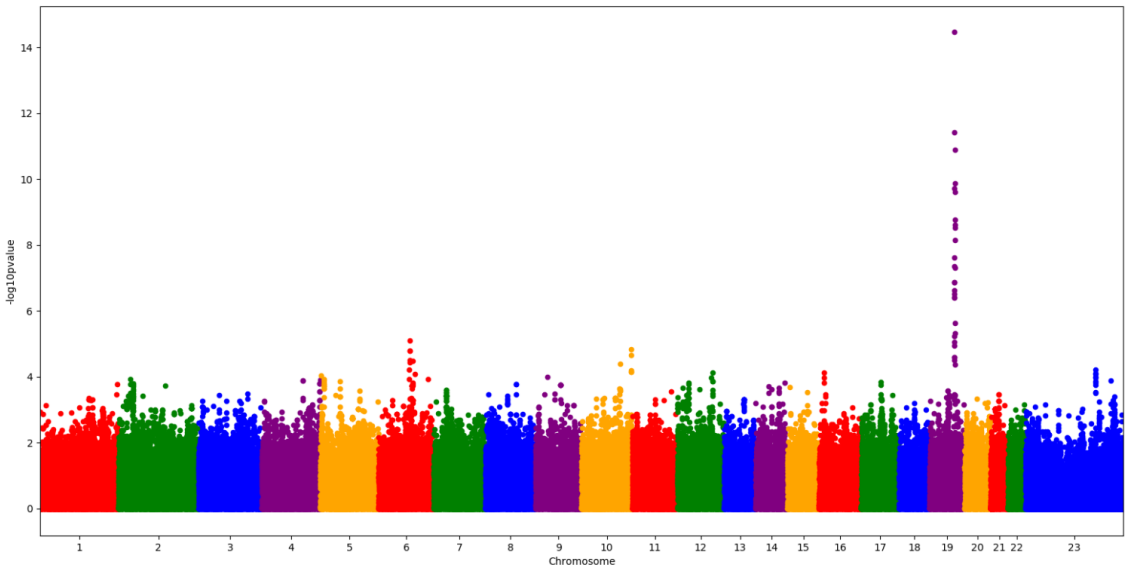
W celu poprawy jakości klasyfikacji oraz zmniejszenia przetrenowania modelu sprawdzono jaki wpływ na dokładność klasyfikacji ma wybór jedynie najistotniejszych (z punktu widzenia klasyfikatora lasów losowych) SNP-ów. Stworzono wiele modeli, z których każdy opiera się na innej części najistotniejszych SNP-ów ($5, 10, 15, ..., 95, 100$ % najlepszych). Dokładność każdego modelu oraz odchylenie standardowe wyniku przedstawiono na wykresie 5.1. Jak widać, istotny wzrost dokładności klasyfikacji jest obserwowany do momentu uwzględnienia około 30% najlepszych SNP-ów. Dalsze poszerzanie zbioru atrybutów nie podnosi znacząco (biorąc pod uwagę odchylenia standardowe wyników) jakości klasyfikacji, co sugeruje, że 30% najistotniejszych SNP-ów niesie prawie całą informacje na temat pacjenta, potrzebną do postawienia diagnozy.



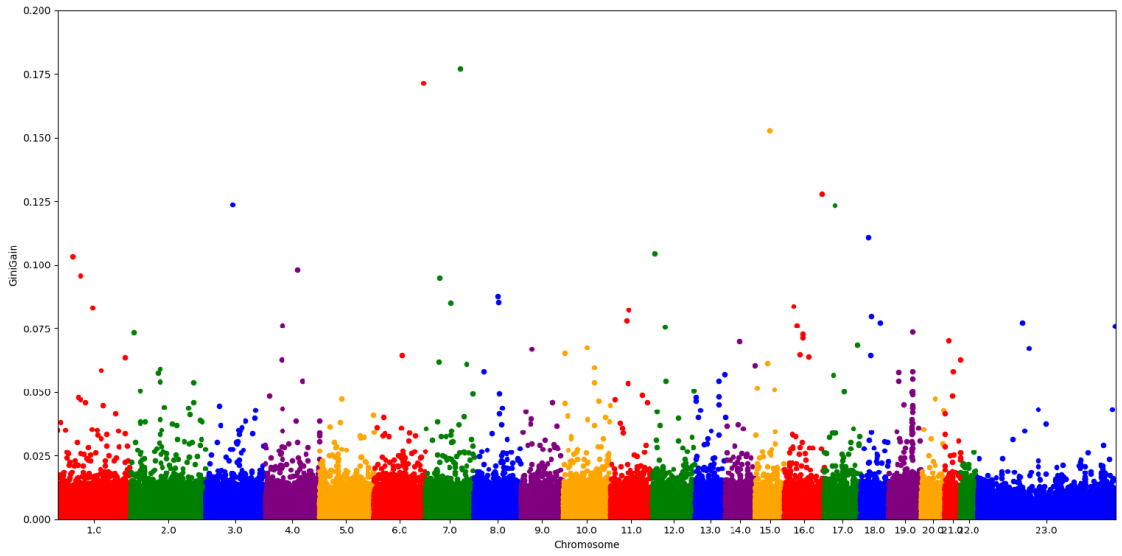
Wykres 5.1: Dokładność klasyfikacji a procent najlepszych SNP-ów wykorzystanych w modelu. Powyższy wykres przedstawia zależność pomiędzy liczbą SNP-ów o największej wartości parametru *ważność* (wykorzystanych do budowy klasyfikatora) a jakością jego predykcji. Parametr *ważność* mówi o istotności danego atrybutu przy tworzeniu klasyfikatora (zobacz sekcję 3.2.2). Dokładność CV jest szacowana na podstawie 10 powtórzeń 10-krotnej walidacji krzyżowej (zobacz sekcję 3.3.2). Pionowe linie oznaczają wartość odchylenia standardowego danego wyniku.

5.2. Analiza najistotniejszych miejsc w genomie

Analizę wybranych przez Borutę SNP-ów rozpoczęto od oceny każdego z nich poza kontekstem pozostałych (inaczej niż przy ocenie na podstawie lasu losowego). Wykorzystano w tym celu dwie miary istotności: *Gini Gain* oraz *p-wartość* (zobacz sekcję 3.1). Wyniki przedstawiono na tak zwanym *Manhattan plot* (wykresy 5.2 i 5.3), czyli wykresie przedstawiającym zależność pomiędzy położeniem danego SNP-a na chromosomie a przypisaną do niego wartością. W przypadku oceny SNP-ów na podstawie *p-wartości* pozycja punktu odpowiadającego danemu polimorfizmowi wzdłuż osi Y zależy od wartości wyrażenia $-\log_{10} pvalue$. W ten sposób im mniejsza *p-wartość*, tym wyżej jest położony dany SNP. Warto w tym miejscu przypomnieć, że *p-wartość* jest wyliczana na podstawie testu χ^2, dla hipotezy zerowej mówiącej, że dany SNP nie ma żadnego wpływu na diagnozę (im mniejsza *p-wartość* tym jest to mniej prawdopodobne). Dzięki temu szczególnie istotne miejsca w genomie, charakteryzujące się obecnością SNP-ów wysoce skorelowanych z występowaniem choroby, uwidaczniają się w postaci pionowych linii. Na podstawie wykresu 5.2 można stwierdzić, że szczególnie istotne miejsca zlokalizowane są na chromosomie 6 i 19. Wykres 4.3, prezentujący wartości *Gini Gain*, nie umożliwia łatwej lokalizacji wyróżniających się miejsc w genomie, choć na chromosomie 19 wciąż widać obszar o szczególnym zagęszczeniu ważnych SNP-ów.



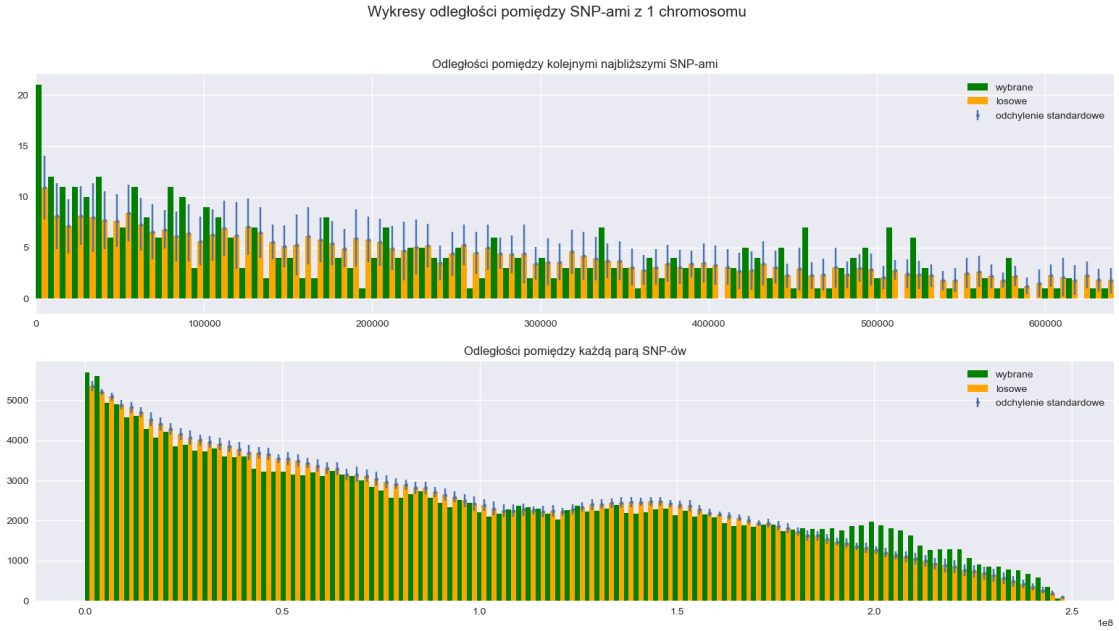
Wykres 5.2: Manhattan plot dla *p-wartości*
Wykres stworzony na podstawie wybranych przez Borutę SNP-ów (z parametrem `perc`=95). Każdy punkt to wartość $-\log_{10} pvalue$, gdzie *pvalue* to *p-wartość* wyliczona na podstawie jednostronnego testu χ^2. Kolorami wyodrębniono poszczególne chromosomy.



Wykres 5.3: Manhattan plot dla wartości *Gini Gain*
Wykres stworzony na podstawie wybranych przez Borutę SNP-ów (z parametrem `perc`=95). Każdy punkt to wartość *Gini Gain* (informacja o ile zmniejszyła się *Gini impurity* po podziale zbioru danych według danego SNP-a - zobacz sekcja 3.1.1). Kolorami wyodrębniono poszczególne chromosomy.

5.2.1. Analiza odległości

Analiza odległości pomiędzy polimorfizmami uznanymi za najistotniejsze dostarcza dodatkowych informacji na temat ich rozłożenia w genomie. Można się spodziewać, że SNP-y mające znaczenie dla rozwoju choroby będą grupować się wokół istotnych genów bądź obszarów regulatorowych. To oznacza, że rozkład odległości pomiędzy kolejnymi SNP-ami bądź też pomiędzy każdą możliwą parą SNP-ów powinien być wzbogacony w wartości małe (odległości pomiędzy SNP-ami związanymi z jednym obszarem o wspólnej funkcji lub wysokim poziomem sprzężenia genetycznego) oraz w wartości duże (odległości pomiędzy SNP-ami z różnych grup/obszarów). Na wykresie 5.4 przedstawiono porównanie rozkładu odległości między najistotniejszymi SNP-ami oraz między SNP-ami wybranymi losowo z 1 chromosomu. Odległości pomiędzy tymi uznanymi za najistotniejsze rzeczywiście rozkładają się mniej równomiernie. Na histogramie pokazującym rozkład odległości między kolejnymi SNP-ami (górny panel wykresu 5.4) dla tych najistotniejszych wyraźnie widoczne jest nagromadzenie odległości małych. Z kolei na histogramie pokazującym rozkład odległości między każdą możliwą parą SNP-ów (dolny panel wykresu 5.4) widoczna jest większa liczba odległości dużych.



Wykres 5.4: Wykresy odległości dla chromosomu 1
Górny histogram: odległości w genomie pomiędzy kolejnymi SNP-ami. Zielone słupki odpowiadają odległościom pomiędzy wybranymi przez Borutę SNP-ami, podczas gdy pomarańczowe to odległości pomiędzy losowo wybranymi SNP-ami z tego samego chromosomu. Dla losowych wartości obliczono również odchylenia standardowe: zaznaczone niebieską linią (szacowanie na podstawie 20 powtórzeń).
Dolny histogram: odległości w genomie liczone pomiędzy każdą parą SNP-ów. Słupki koloru zielonego przedstawiają odległości wyliczone na podstawie wybranych przez Borutę SNP-ów, z kolei pomarańczowe odpowiadają SNP-om wybranym losowo. Niebieską linią zaznaczono odchylenia standardowe (szacowanie na podstawie 10 powtórzeń).

5.2.2. Porównanie do badań GWAS

Wybrane za pomocą algorytmu Boruta SNP-y porównano do SNP-ów zidentyfikowanych jako powiązane z chorobą Alzheimera w trakcie badań typu GWAS (zobacz 1.2). W tym celu wykorzystano metodę `intersect` z pakietu narzędzi do analizy porównawczej genomów *bedtools* (`bedtools.readthedocs.io`) oraz listę SNP-ów pochodzących z analiz GWAS [20]. Pozycje tych SNP-ów porównano do fragmentów genomu wyodrębnionych na podstawie lokalizacji SNP-ów wybranych przez Borutę (przy `perc`=95) rozszerzonych o ± 100 par zasad. W tabeli 5.2 przedstawiono listę SNP-ów wspólnych dla wyników analiz GWAS oraz opisanej w niniejszej pracy selekcji cech za pomocą metody Boruta.

Tabela 5.2: W tabeli przedstawiono SNP-y wyodrębnione jako istotne w procesie rozwoju choroby Alzheimera zarówno podczas badań GWAS [20] jak i podczas analizy algorytmem Boruta z parametrem `perc`=95. Kolumna **pozycja** mówi o położeniu danego SNP-a (łatwego do identyfikacji poprzez **numer SNP-a**) na chromosomie. **Nazwa genu** mówi o genie, na który najprawdopodobniej ma wpływ dany SNP.

chromosom	**pozycja danego SNP-a**	**numer SNP-a**	**nazwa genu**
2	227562139	rs7558386	IRS1, MIR5702
4	2103096	rs1923775	POLN
5	118435127	rs116348108	DMXL1
9	78793421	rs11144781	PCSK5
11	85831541	rs471470	PICALM, FNTAP1
11	113106455	rs12279261	NCAM1
11	125193596	rs11601321	PKNOX2
19	45392254	rs6857	PVRL2
19	45395619	rs2075650	C1RL-AS1
19	45396219	rs157582	TOMM40
19	45396665	rs59007384	TOMM40
19	45410002	rs769449	APOE
19	45411941	rs429358	APOE
19	45422846	rs56131196	APOC1, APOC1P1
19	45422946	rs4420638	APOC1, APOC1P1
21	14928355	rs9975691	LOC102724188

Wspólnych SNP-ów jest mało w porównaniu do wszystkich uznanych za istotne (wszystkich SNP-ów wybranych przez Borutę z parametrem `perc`=95 jest 129 214 - zobacz tabelę 5.1). Może wskazywać to na małą uniwersalność stworzonego modelu, gdyż nie wykorzystuje on wielu z najpowszechniejszych polimorfizmów związanych z chorobą Alzheimera. Z drugiej strony wynik potwierdza (podobnie jak wykres 5.2) istotność rejonów w obrębie 19 chromosomu (między innymi tych, związanych z genem APOE) w rozwoju AD.

Rozdział 6

Dyskusja

Genetyczne podstawy choroby Alzheimera nie są w pełni poznane. Do tej pory opisano jednak wiele zmian genetycznych powiązanych z chorobą. Należą do nich mutacje w genach związanych z metabolizmem białka APP (geny APP, PSEN1, PSEN2) i cholesterolu (na przykład gen APOE) oraz w tych, mających związek z odpowiedzią immunologiczną (między innymi gen CR1) oraz z procesem endocytozy (na przykład gen SORL1). Celem niniejszy pracy było stworzenie klasyfikatora pozwalającego na odróżnienie osób zdrowych od chorych na chorobę Alzheimera na podstawie danych genetycznych. Równie ważnym aspektem tej pracy było późniejsze wskazanie polimorfizmów genetycznych najistotniejszych z punktu widzenia procesu klasyfikacji. Największą trudność stanowił wybór najważniejszych SNP-ów, wykorzystanych następnie jako atrybuty w trakcie budowy opisanego wyżej modelu. W tym celu wykorzystano algorytm Boruta oraz metody drzewiaste. Wynikiem było kilka różnej wielkości zbiorów najważniejszych SNP-ów oraz klasyfikatory zbudowane na ich podstawie. Najwyższą dokładność osiągnął klasyfikator lasów losowych, który poprawnie przypisał diagnozę do ponad 98% obiektów ze zbioru testowego.

Wcześniejsze prace na temat klasyfikacji pacjentów chorych na Alzheimera na podstawie danych genetycznych w większości analizują dane z badań typu GWAS, uzyskanych w oparciu o macierze genotypujące [21]-[26]. Wykorzystują w tym celu głównie metody drzewiaste, ale również regresję logistyczną czy maszyny wektorów wspierających [24], uzyskując dokładność klasyfikacji w zakresie od 50% [21] aż do 90% [22]. Uzyskana w niniejszej pracy wysoka jakość klasyfikacji może sugerować, że wykorzystanie danych pochodzących z sekwencjonowania całogenomowego (WGS - zobacz 1.2) dostarcza dodatkowych, wartościowych informacji, których brak w danych GWAS. Jednak biorąc pod uwagę szacunki, że geny stanowią około 50-80% czynnika ryzyka zachorowania na Alzheimera, jest wysoce prawdopodobne, że osiągnięta dokładność jest rezultatem przetrenowania modelu. Pośrednio może na to wskazywać duża różnica między wynikami uzyskanymi przez klasyfikator lasów losowych a wynikami dla drzewa decyzyjnego (zobacz tabelę 5.1). Warto zauważyć, że oba klasyfikatory zostały zbudowane na zestawie atrybutów wybranych przez algorytm Boruta, który oceniał ich istotność na podstawie przydatności do klasyfikacji w lesie losowym. Można więc podejrzewać, że Boruta wybiera z danych wejściowych te SNP-y, które najlepiej sprawdzą się właśnie w tego typu klasyfikatorze. Słabszy wynik drzewa decyzyjnego opartego na tych samych danych sugeruje, że atrybuty najlepsze z punktu widzenia lasu losowego nie są uniwersalne dla każdego modelu, więc ich istotność w rozwoju choroby Alzheimera nie jest oczywista. Z drugiej strony porównując wyniki drzew decyzyjnych zbudowanych na podstawie wybranych przez Borutę SNP-ów do wyników uzyskanych za pomocą drzew decyzyjnych na podstawie SNP-ów pochodzących z

analiz GWAS widać, że te pierwsze osiągają wynik lepszy o około 10% [21].

Porównanie zestawu SNP-ów wybranych za pomocą algorytmu Boruta do wyników analiz typu GWAS pokazało, że podejścia te dają dość odrębne wyniki. Wspólnych SNP-ów jest mało w porównaniu do wszystkich uznanych za istotne przez algorytm Boruta (wszystkich SNP-ów wybranych przez Borutę z parametrem `perc`=95 jest 129 214 - zobacz tabelę 5.1, wspólnych natomiast kilkanaście). Wynik ten może być efektem przeprowadzenia selekcji atrybutów na podzbiorach SNP-ów. Może również wskazywać na małą uniwersalność stworzonego modelu, gdyż nie wykorzystuje on wielu z najpowszechniejszych polimorfizmów związanych z chorobą Alzheimera. Z drugiej strony wynik potwierdza (podobnie jak wykres 5.2) istotność rejonów w obrębie chromosomu 19 (między innymi tych, związanych z genem APOE) w rozwoju AD.

Podsumowując, uzyskane wyniki sugerują, że zastosowanie metod selekcji cech i klasyfikatora lasów losowych do danych pochodzących z WGS może pozwolić zidentyfikować nowe genetyczne czynniki ryzyka, związane z chorobą Alzheimera i stanowić wartość dodaną w stosunku do analiz GWAS, jednak stwierdzenie tego faktu wymaga dalszych badań.

Bibliografia

[1] World Health Organization, *Dementia Fact sheet*, 12 Dec 2017

[2] Gaël Nicolas, Camille Charbonnier, Dominique Campion (2016): *From Common to Rare Variants: The Genetic Component of Alzheimer Disease*, Human Heredity, 81:129–141

[3] Bloom GS (2014): *Amyloid-β and tau: the trigger and bullet in Alzheimer disease pathogenesis*, JAMA Neurology, 71(4):505-8

[4] Chávez-Gutiérrez L, Bammens L, Benilova I, Vandersteen A, Benurwar M, Borgers M, Lismont S, Zhou L, Van Cleynenbreugel S, Esselmann H, Wiltfang J, Serneels L, Karran E, Gijsen H, Schymkowitz J, Rousseau F, Broersen K, De Strooper B. (2012): *The mechanism of γ-Secretase dysfunction in familial Alzheimer disease*, EMBO Journal, 31(10):2261-74

[5] Don W.Cleveland, Shu-Ying Hwo, Marc W.Kirschner (1977): *Physical and chemical properties of purified tau factor and the role of tau in microtubule assembly*, Journal of Molecular Biology, Volume 116, Issue 2, Pages 227-247

[6] Mudher A, Lovestone S. (2002): *Alzheimer's disease – do tauists and baptists finally shake hands?*, Trends in Neurosciences, 25(1):22-6

[7] Guerreiro R, Brás J, Hardy J (2013): *SnapShot: genetics of Alzheimer's disease.*, Cell, 155(4):968-968

[8] Jean-Charles Lambert, Carla A Ibrahim-Verbaas, Denise Harold, Adam C Naj, Rebecca Sims at al. (2013): *Meta-analysis of 74,046 individuals identifies 11 new susceptibility loci for Alzheimer's disease*, Nature Genetics, 45(12):1452–1458

[9] Seshadri S, Fitzpatrick AL, Ikram MA, DeStefano AL, Gudnason V et al. (2010): *Genome-wide analysis of genetic loci associated with Alzheimer disease*, JAMA, 303(18):1832-40

[10] Caroline Van Cauwenberghe, Christine Van Broeckhoven, Kristel Sleegers (2016): *The genetic landscape of Alzheimer disease: clinical implications and perspectives*, Genetics in Medicine, 18(5):421–430

[11] Robert S. Wilson, Sandra Barral, Joseph H. Lee, Sue E. Leurgans, Tatiana M. Foroud, Robert A. Sweet, Neill Graff-Radford, Thomas D. Bird, Richard Mayeux, David A. Bennett (2011): *Heritability of Different Forms of Memory in the Late Onset Alzheimer's Disease Family Study*, Journal of Alzheimer's disease, 23(2):249–255

[12] Celeste M. Karch, Alison M. Goate (2015): *Alzheimer's disease risk genes and mechanisms of disease pathogenesis*, Biological Psychiatry, 77(1):43-51

[13] Kim J, Basak JM, Holtzman DM (2009): *The role of apolipoprotein E in Alzheimer's disease*, Neuron, 63:287–303

[14] Heppner FL, Ransohoff RM, Becher B (2015): *Immune attack: the role of inflammation in Alzheimer disease*, Nature reviews. Neuroscience, 16(6):358-72

[15] Holtzman DM, Morris JC, Goate AM (2011): *Alzheimer's disease: The challenge of the second century*, Science Translational Medicine, Volume 3, Issue 77, pp. 77sr1

[16] Liu D, Niu ZX (2009): *The structure, genetic polymorphisms, expression and biological functions of complement receptor type 1 (CR1/CD35)*, Immunopharmacol Immunotoxicol, 31:524–535

[17] Miron B. Kursa, Witold R. Rudnicki (2010): *Feature Selection with the Boruta Package*, Journal of Statistical Software, Volume 36, Issue 11

[18] Petr Danecek, Adam Auton, Goncalo Abecasis, Cornelis A. Albers, Eric Banks, Mark A. DePristo, Robert E. Handsaker, Gerton Lunter, Gabor T. Marth, Stephen T. Sherry, Gilean McVean, Richard Durbin, 1000 Genomes Project Analysis Group (2011): *The variant call format and VCFtools*, Bioinformatics, Volume 27, Issue 15, 2156–2158

[19] Aaron McKenna, Matthew Hanna, Eric Banks, Andrey Sivachenko, Kristian Cibulskis, Andrew Kernytsky, Kiran Garimella, David Altshuler, Stacey Gabriel, Mark Daly, Mark A. DePristo (2010): *The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data*, Genome Research, 20:1297-1303

[20] Sanila Amber, Saadia Zahid (2018): *Data integration for functional annotation of regulatory single nucleotide polymorphisms associated with Alzheimer's disease susceptibility*, Gene, 672:115-125

[21] Onur Erdogan, Yesim Aydin Son (2014): *Predicting the Disease of Alzheimer With SNP Biomarkers and Clinical Data Using Data Mining Classification Approach: Decision Tree*, Studies in Health Technology and Informatics, 205:511-515

[22] Thanh-Tung Nguyen, Joshua Zhexue Huang, Qingyao Wu, Thuy Thi Nguyen, Mark Junjie Li (2015): *Genome-wide association data classification and SNPs selection using two-stage quality-based Random Forests*, BMC Genomics, 16(Suppl 2):S5

[23] Natalia Briones, Valentin Dinu (2012): *Data mining of high density genomic variant data for prediction of Alzheimer's disease risk*, BMC Medical Genetics, 13:7

[24] Xia Jiang, Binghuang Cai, Diyang Xue, Xinghua Lu, Gregory F Cooper, Richard E Neapolitan (2014): *A comparative analysis of methods for predicting clinical outcomes using high-dimensional genomic datasets*, Journal of the American Medical Informatics Association, Volume 21, Issue e2, Pages e312–e319

[25] Fayroz F. Sherif, Nourhan Zayed, Mahmoud Fakhr, Manal Abdel Wahed, Yasser M. Kadah (2017): *Integrated Higher-Order Evidence-Based Framework for Prediction of Higher-Order Epistasis Interactions in Alzheimer's Disease*, International Journal of Biology and Biomedical Engineering, Volume 11

[26] Matthew E Stokes, M Michael Barmada, M Ilyas Kamboh, Shyam Visweswaran (2014): *The application of network label propagation to rank biomarkers in genome-wide Alzheimer's data*, BMC Genomics, 15:282